

TUGAS AKHIR - KI091391

IMPLEMENTASI FUZZY LOGIC BASED RANKING FUNCTION DALAM SISTEM INFORMASI PENCARIAN ARTIKEL PARIWISATA BERBASIS MOBILE

MARVIN ZESON ABILO
NRP 5112 100 051

Dosen Pembimbing
Diana Purwitasari, S.Kom., M.Sc.
Rizky Januar Akbar, S.Kom., M.Eng.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016



TUGAS AKHIR – KI091391

**IMPLEMENTASI FUZZY LOGIC BASED
RANKING FUNCTION DALAM SISTEM
INFORMASI Pencarian ARTIKEL
PARIWISATA BERBASIS MOBILE**

MARVIN ZESON ABILO
NRP 5112 100 051

Dosen Pembimbing
Diana Purwitasari, S.Kom., M.Sc.
Rizky Januar Akbar, S.Kom., M.Eng.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016

[Halaman ini sengaja dikosongkan]



FINAL PROJECT – KI091391

IMPLEMENTATION OF FUZZY LOGIC BASED RANKING FUNCTION FOR MOBILE BASED TOURISM INFORMATION RETRIEVAL SYSTEM

MARVIN ZESON ABILO
NRP 5112 100 051

Advisor
Diana Purwitasari, S.Kom., M.Sc.
Rizky Januar Akbar, S.Kom., M.Eng.

INFORMATICS DEPARTMENT
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2016

[Halaman ini sengaja dikosongkan]

**IMPLEMENTASI FUZZY LOGIC BASED RANKING
FUNCTION DALAM SISTEM INFORMASI PENCARIAN
ARTIKEL PARIWISATA BERBASIS MOBILE**

TUGAS AKHIR

**Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Komputasi Cerdas dan Visi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember**

**Oleh :
MARVIN ZESON ABILO
NRP : 5112 100 051**

Disetujui oleh Dosen Pembimbing Tugas Akhir

**Diana Purwitasari, S.Kom., M.Si
NIP: 197804102003122001**



**Rizky Januar Akbar, S.Kom., M.Eng.
NIP: 198701032014041001**

**SURABAYA
JULI 2016**

[Halaman ini sengaja dikosongkan]

**IMPLEMENTASI FUZZY LOGIC BASED RANKING
FUNCTION DALAM SISTEM INFORMASI Pencarian
ARTIKEL PARIWISATA BERBASIS MOBILE**

Nama Mahasiswa : MARVIN ZESON ABILO
NRP : 5112 100 051
Jurusan : Teknik Informatika ITS
Dosen Pembimbing I : Diana Purwitasari, S.Kom., M.Sc.
Dosen Pembimbing II : Rizky Januar Akbar, S.Kom., M.Eng.

Abstrak

Kebutuhan manusia terhadap perkembangan teknologi informasi semakin meningkat. Begitu pula dengan kebutuhan manusia untuk mendapatkan sebuah informasi dengan cepat. Sebuah informasi didapatkan dengan menggunakan sebuah sistem temu kembali informasi yang mengembalikan sejumlah dokumen relevan dari sebuah dataset yang besar sesuai dengan query masukan pengguna. Pengembalian informasi relevan ini dilakukan berdasarkan hasil pembobotan dokumen dengan menggunakan metode tertentu. Metode paling umum yang digunakan dalam pembobotan dokumen adalah pembobotan TF – IDF(term frequency-inverse document frequency) yang merupakan hasil perkalian term frequency dengan inverse document frequency.

Pada Tugas Akhir ini akan diimplementasikan sebuah metode pembobotan dokumen yang menggunakan pendekatan fuzzy untuk meningkatkan akurasi dari sebuah sistem temu kembali informasi. Metode tersebut akan menggunakan beberapa elemen yang sering digunakan dalam pembobotan dokumen, seperti term frequency, inverse document frequency dan normalization. Metode ini akan melakukan pendekatan fuzzy dengan 2 level untuk menghitung nilai relevansi dari sebuah dokumen.

Hasil uji coba menunjukkan metode dengan pendekatan fuzzy menghasilkan nilai precision dan recall yang lebih baik jika dibandingkan dengan metode non fuzzy.

Kata kunci: fuzzy logic, sistem temu kembali informasi, ranking function, pariwisata, mobile.

IMPLEMENTATION OF FUZZY LOGIC BASED RANKING FUNCTION FOR MOBILE BASED TOURISM INFORMATION RETRIEVAL SYSTEM

Name : MARVIN ZESON ABILO
NRP : 5112 100 051
Major : Informatics Engineering Department – ITS
Supervisor I : Diana Purwitasari, S.Kom., M.Sc.
Supervisor II : Rizky Januar Akbar, S.Kom., M.Eng.

Abstract

Human needs to the development of information technology is increasing. Similarly, the human need to obtain an information quickly. Information obtained using an information retrieval system that returns a number of relevant documents from a large dataset in accordance with the user input query. Relevant information retrieval is done based on the weighting of the document by using certain methods. The most common method used in weighting the document is TF – IDF (term frequency-inverse document frequency) weighting which is the result of the multiplication term frequency by inverse document frequency.

In this final project will be implemented a document weighting method that uses fuzzy approach to improve the accuracy of an information retrieval system. The method uses several elements that are often used in weighting the document, such as term frequency, inverse document frequency and normalization. This method will perform two level fuzzy approach to calculate the relevance of a document

Experiment results show the method with fuzzy approach generate better precision and recall value when compared to non-fuzzy method.

Keywords: fuzzy logic, information retrieval system, ranking function, tourism, mobile.

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Segala puji dan syukur ke hadirat Tuhan yang Maha Esa yang telah memberikan berkat dan tuntunan-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “*Implementasi Fuzzy Logic Based Ranking Function dalam Sistem Informasi Pencarian Artikel Pariwisata Berbasis Mobile*”.

Dalam pelaksanaan Tugas Akhir ini tentu penulis sebagai makhluk sosial tidak dapat menyelesaikannya tanpa bantuan dari pihak lain. Tanpa mengurangi rasa hormat, penulis memberikan penghargaan serta ucapan terima kasih yang sebesar-besarnya kepada:

1. Orang tua, saudara serta keluarga besar yang senantiasa memberikan semangat dan doa agar penulis dapat menyelesaikan Tugas Akhir dengan tepat waktu.
2. Ibu Diana Purwitasari, S.Kom, M.Sc selaku dosen pembimbing Tugas Akhir pertama yang telah membimbing, memotivasi dan memberikan banyak masukan dalam pengerjaan Tugas Akhir ini.
3. Bapak Rizky Januar Akbar, S.Kom., M.Eng. selaku dosen pembimbing Tugas Akhir kedua yang selalu memberikan masukan-masukan yang dapat penulis kembangkan dari Tugas Akhir ini.
4. Bapak Daniel Oranova Siahaan, S.Kom, M.sc., PDEng selaku dosen wali selama penulis menjalani proses perkuliahan.
5. Bapak dan Ibu dosen Jurusan Teknik Informatika ITS yang telah mengajarkan banyak ilmu berharga kepada penulis.
6. Bapak dan Ibu karyawan Jurusan Teknik Informatika ITS atas berbagai bantuan yang telah diberikan kepada penulis selama masa perkuliahan.

7. Teman-teman administrator Laboratorium Komputasi Cerdas dan Visi yang telah memberi banyak masukan selama Tugas Akhir ini
8. Teman-teman Teknik Informatika ITS angkatan 2012, yang telah memberikan warna-warni kehidupan mahasiswa mulai sejak mahasiswa baru hingga lulus.
9. Pihak-pihak lain yang tidak sempat penulis sebutkan, yang telah membantu kelancaran pengerjaan Tugas Akhir ini.

Penulis sangat berharap bahwa apa yang dihasilkan dari Tugas Akhir ini bisa memberikan manfaat bagi semua pihak, khususnya bagi diri penulis sendiri dan seluruh *civitas academica* Teknik Informatika ITS, serta bagi agama, bangsa, dan negara. Tak ada manusia yang sempurna sekalipun penulis berusaha sebaik mungkin dalam menyelesaikan Tugas Akhir ini. Karena itu, penulis memohon maaf apabila terdapat kesalahan, kekurangan, maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun sangat diharapkan oleh penulis untuk dapat disampaikan untuk perbaikan selanjutnya.

Surabaya, Juli 2016

Marvin Zeson Abilo

DAFTAR ISI

Abstrak	vii
Abstract.....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL	xxi
DAFTAR KODE SUMBER	xxiii
1 BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	4
1.3. Batasan Masalah	4
1.4. Tujuan	4
1.5. Metodologi	5
1.6. Sistematika Penulisan	6
2 BAB II TINJAUAN PUSTAKA.....	9
2.1. Pengambilan Data Menggunakan Import.io	9
2.2. Preprocessing Data.....	11
2.3. Pembobotan Dokumen TF – IDF	12
2.4. Pembobotan Dokumen Dengan Pendekatan Fuzzy	13
2.5. Komunikasi Web Service dengan Android.....	21
3 BAB III ANALISIS DAN PERANCANGAN.....	25
3.1. Analisis Implementasi Secara Umum	25
3.2. Perancangan Data.....	27

3.2.1.	Data Masukan	27
3.2.2.	Data Proses.....	27
3.2.3.	Data Keluaran	29
3.3.	Perancangan Proses.....	31
3.3.1.	Tahap Pengumpulan dan Persiapan Data	31
3.3.2.	Tahap Perancangan Pembobotan Dokumen dengan Pendekatan Fuzzy	39
3.3.3.	Tahap Perancangan Web Service.....	60
3.3.4.	Tahap Perancangan Aplikasi Mobile	70
3.4.	Perancangan Antarmuka Perangkat Lunak	72
3.4.1.	Slide Drawer Navigasi	72
3.4.2.	Halaman Utama.....	74
3.4.3.	Halaman Pemilihan Kategori	75
3.4.4.	Halaman Pencarian	76
3.4.5.	Halaman Detail Artikel	77
4	BAB IV IMPLEMENTASI	79
4.1.	Lingkungan Implementasi.....	79
4.2.	Implementasi Proses	79
4.2.1.	Implementasi Tahap Pengumpulan dan Persiapan Data	81
4.2.2.	Implementasi Tahap Perancangan Proses Pembobotan Dokumen dengan Pendekatan Fuzzy	86
4.2.3.	Implementasi Tahap Perancangan Web Service	97
4.2.4.	Implementasi Tahap Perancangan Aplikasi Mobile	102
4.3.	Implementasi Antarmuka Pengguna	104
4.3.1.	Slide Drawer Navigasi	104

4.3.2.	Halaman Utama.....	105
4.3.3.	Halaman Pemilihan Kategori	106
4.3.4.	Halaman Pencarian	107
4.3.5.	Halaman Detail Artikel	108
5	BAB V UJI COBA DAN EVALUASI	111
5.1.	Lingkungan Uji Coba.....	111
5.2.	Data Uji Coba	111
5.3.	Skenario Uji Coba.....	112
5.4.	Skenario Pengujian 1: Perbandingan Penggunaan Parameter Norm Terhadap Hasil Akurasi Pencarian Artikel	113
5.5.	Skenario Pengujian 2: Perbandingan Bentuk Rule Fuzzy Terhadap Hasil Akurasi Pencarian Artikel	118
5.6.	Skenario Pengujian 3: Perbandingan Penggunaan FLC Query Terhadap Hasil Akurasi Pencarian Artikel	123
5.7.	Skenario Pengujian 4: Perbandingan Pendekatan Fuzzy dan Non Fuzzy Terhadap Hasil Akurasi Pencarian Artikel ..	127
5.8.	Analisa Uji Coba.....	137
6	BAB VI KESIMPULAN DAN SARAN	139
6.1.	Kesimpulan	139
6.2.	Saran	139
7	DAFTAR PUSTAKA.....	141
8	LAMPIRAN 1: Modul Sistem.....	143
9	LAMPIRAN 2: Daftar Query Uji Coba dan Artikel Relevan.....	149
10	BIODATA PENULIS.....	159

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 1.1. Hasil search tidak relevan pada website wisataindonesia.co.id	3
Gambar 1.2. Hasil search relevan pada website wisataindonesia.co.id	3
Gambar 2.1. Diagram Aktivitas Penggunaan Import.io.....	10
Gambar 2.2. Contoh Membership Function Parameter untuk Pembobotan Dokumen (1)	14
Gambar 2.3. Contoh Membership Function Parameter untuk Pembobotan Dokumen (2)	15
Gambar 2.4. Struktur Fuzzy	17
Gambar 2.5. Hubungan Antar Parameter Dalam Fuzzy [12].....	18
Gambar 2.6. Diagram Alir Penggunaan KSOAP2.....	24
Gambar 3.1. Diagram Alir Implementasi Program	26
Gambar 3.2. Data Keluaran di Perangkat Mobile	30
Gambar 3.3. Arsitektur Sistem Pencarian Artikel.....	31
Gambar 3.4. Diagram Alir Tahap Pengumpulan Data	32
Gambar 3.5. Proses Training Pada Import.io	33
Gambar 3.6. Struktur Objek Artikel.....	34
Gambar 3.7. Hasil Crawling Menggunakan Import.io.....	35
Gambar 3.8. Diagram Alir Penggunaan Library GSON	36
Gambar 3.9. Diagram Kelas Program Parsing Data JSON ke Objek Artikel	37
Gambar 3.10. Diagram Alir Proses Stopword Removal	38
Gambar 3.11. Diagram Alir Tahap Perancangan Fuzzy	39
Gambar 3.12. Diagram Alir Proses Perancangan Membership Function	40
Gambar 3.13. Aturan Penentuan Membership Function.....	44
Gambar 3.14. Membership Function Term Frequency Didalam Query	49
Gambar 3.15. Membership Function Inverse Document Frequency	52
Gambar 3.16. Grafik Perbandingan Nilai Inverse Query Length dengan Panjang Query	55

Gambar 3.17. Membership Function Inverse Query Length	57
Gambar 3.18. Contoh Proses Penggabungan Rule.....	58
Gambar 3.19. Diagram Alir Tahap Perancangan Web Service ..	62
Gambar 3.20. Diagram Alir Berjalannya Web Service.....	63
Gambar 3.21. Diagram Kelas Web Service	64
Gambar 3.22. Contoh Skema Request Web Service.....	68
Gambar 3.23. Contoh Skema Response Web Service	69
Gambar 3.24. Diagram Kelas Aplikasi Mobile.....	70
Gambar 3.25. Rancangan Slide Drawer Navigasi.....	73
Gambar 3.26. Rancangan Halaman Utama Aplikasi	74
Gambar 3.27. Rancangan Halaman Pemilihan Kategori dan Halaman yang Menampilkan Artikel Berdasar Kategori	75
Gambar 3.28. Rancangan Halaman Pencarian.....	77
Gambar 3.29. Halaman Baca Artikel	78
Gambar 4.1. Penentuan Atribut yang Akan Diambil	80
Gambar 4.2. Penentuan Elemen yang Akan Diambil Untuk Atribut Judul.....	81
Gambar 4.3. Penentuan Elemen yang Akan Diambil Untuk Atribut Isi	82
Gambar 4.4. Tampilan Konfigurasi Crawler	82
Gambar 5.1. Nilai Precision Pada 10 Dokumen Pertama Berdasarkan 11 Data Uji Coba.....	115
Gambar 5.2. Nilai Precision Pada 20 Dokumen Pertama Berdasarkan 11 Data Uji Coba.....	116
Gambar 5.3. Nilai Precision Pada 30 Dokumen Pertama Berdasarkan 11 Data Uji Coba.....	116
Gambar 5.4. Nilai Recall Pada 10 Dokumen Pertama Berdasarkan 11 Data Uji Coba	117
Gambar 5.5. Nilai Recall Pada 20 Dokumen Pertama Berdasarkan 11 Data Uji Coba	117
Gambar 5.6. Nilai Recall Pada 30 Dokumen Pertama Berdasarkan 11 Data Uji Coba	118
Gambar 5.7. Nilai Precision Pada 10 Dokumen Pertama Berdasarkan 11 Data Uji Coba.....	120

Gambar 5.8. Nilai Precision Pada 20 Dokumen Pertama Berdasarkan 11 Data Uji Coba.....	120
Gambar 5.9. Nilai Precision Pada 30 Dokumen Pertama Berdasarkan 11 Data Uji Coba.....	121
Gambar 5.10. Nilai Recall Pada 10 Dokumen Pertama Berdasarkan 11 Data Uji Coba	121
Gambar 5.11. Nilai Recall Pada 20 Dokumen Pertama Berdasarkan 11 Data Uji Coba	122
Gambar 5.12. Nilai Recall Pada 30 Dokumen Pertama Berdasarkan 11 Data Uji Coba	122
Gambar 5.13. Nilai Precision Pada 10 Dokumen Pertama Berdasarkan 11 Data Uji Coba.....	124
Gambar 5.14. Nilai Precision Pada 20 Dokumen Pertama Berdasarkan 11 Data Uji Coba.....	125
Gambar 5.15. Nilai Precision Pada 30 Dokumen Pertama Berdasarkan 11 Data Uji Coba.....	125
Gambar 5.16. Nilai Recall Pada 10 Dokumen Pertama Berdasarkan 11 Data Uji Coba	126
Gambar 5.17. Nilai Recall Pada 20 Dokumen Pertama Berdasarkan 11 Data Uji Coba	126
Gambar 5.18. Nilai Recall Pada 30 Dokumen Pertama Berdasarkan 11 Data Uji Coba	127
Gambar 5.19. Nilai Precision Pada 10 Dokumen Pertama Berdasarkan 11 Data Uji Coba.....	129
Gambar 5.20. Nilai Precision Pada 20 Dokumen Pertama Berdasarkan 11 Data Uji Coba.....	129
Gambar 5.21. Nilai Precision Pada 30 Dokumen Pertama Berdasarkan 11 Data Uji Coba.....	130
Gambar 5.22. Nilai Recall Pada 10 Dokumen Pertama Berdasarkan 11 Data Uji Coba	130
Gambar 5.23. Nilai Recall Pada 20 Dokumen Pertama Berdasarkan 11 Data Uji Coba	131
Gambar 5.24. Nilai Recall Pada 30 Dokumen Pertama Berdasarkan 11 Data Uji Coba	131

Gambar 9.1. Keseluruhan Modul Sistem Pencarian Artikel Pariwisata.....	143
Gambar 9.2. Isi Package Interface	143
Gambar 9.3. Kelas yang Mengimplementasikan Interface Calculator.....	144
Gambar 9.4. Kelas yang Mengimplementasikan Interface MembershipFunctionFirstLevel.....	144
Gambar 9.5. Kelas yang Mengimplementasikan Interface MembershipFunctionSecondLevel dan Rules	144
Gambar 9.6. Isi Package FullRuleFuzzy.....	145
Gambar 9.7. Isi Package ModifiedFuzzy	146
Gambar 9.8. Isi Package TfidfFuzzy	147
Gambar 9.9. Isi Package TfidfNonFuzzy	148

DAFTAR TABEL

Tabel 2.1. Notasi didalam Struktur Fuzzy	17
Tabel 2.2. Contoh Perhitungan Pembobotan Fuzzy	19
Tabel 3.1. Data Proses	27
Tabel 3.2. Data Keluaran	30
Tabel 3.3. Hasil Analisa Term Frequency didalam Dokumen....	41
Tabel 3.4. Hasil Analisa Inverse Document Frequency	42
Tabel 3.5. Hasil Analisa Term Frequency didalam Query	43
Tabel 3.6. Hasil Analisa Inverse Document Length	43
Tabel 3.7. Hasil Analisa Inverse Query Length	43
Tabel 3.8. Membership Function Term Frequency di dalam Dokumen	46
Tabel 3.9. Membership Function Term Frequency di dalam Query	48
Tabel 3.10. Membership Function Inverse Document Frequency	51
Tabel 3.11. Membership Function Inverse Document Length ...	53
Tabel 3.12. Membership Function Inverse Query Length	55
Tabel 3.13. Jumlah Kemungkinan Fuzzy Rule	57
Tabel 3.14. Rule Tingkat Pertama	58
Tabel 3.15. Rule Tingkat Kedua	60
Tabel 3.16. Skema Request dan Response Web Service	67
Tabel 3.17. Penjelasan Diagram Kelas Aplikasi Mobile	70
Tabel 3.18. Data Inputan Web Service	71
Tabel 3.19. Elemen - Elemen yang Diambil Dari Response Web Service	72
Tabel 3.20. Spesifikasi Atribut Antarmuka Slide Drawer Navigasi	73
Tabel 3.21. Spesifikasi Atribut Antarmuka Halaman Utama Aplikasi	74
Tabel 3.22. Spesifikasi Atribut Antarmuka Halaman Pemilihan Kategori	75
Tabel 3.23. Spesifikasi Atribut Antarmuka Halaman Artikel Berdasar Kategori	76

Tabel 3.24. Spesifikasi Atribut Antarmuka Halaman Pencarian	77
Tabel 3.25. Spesifikasi Atribut Antarmuka Halaman Detail Artikel	78
Tabel 4.1. Lingkungan Implementasi Sistem.....	79
Tabel 5.1. Rule Pada Fuzzy 2 Parameter	113
Tabel 5.2. Perbandingan Rata-Rata Precision dan Recall Antara Fuzzy 3 Parameter dengan Fuzzy 2 Parameter	114
Tabel 5.3. Perbandingan Rata-Rata Precision dan Recall Antara Fuzzy Full Rule dengan Fuzzy Rule Gabungan.....	119
Tabel 5.4. Rata-Rata Precision dan Recall dari Pendekatan yang Menggunakan FLC Query dan Tanpa FLC Query	123
Tabel 5.5. Perbandingan Rata-Rata Precision dan Recall Antara Metode dengan Pendekatan Fuzzy dan Non Fuzzy	128
Tabel 5.6. Hasil Pencarian dengan Pendekatan Fuzzy dan Non Fuzzy pada Query Wisata Bahari Indonesia	132
Tabel 10.1. Daftar Query Uji Coba dan Artikel Relevan.....	149

DAFTAR KODE SUMBER

Kode Sumber 4.1. Kode Untuk Kelas Artikel	83
Kode Sumber 4.2. Kode Untuk Kelas Data	84
Kode Sumber 4.3. Kode Deserializer Artikel	84
Kode Sumber 4.4. Kode Deserializer Data	85
Kode Sumber 4.5. Kode yang Menunjukkan Proses Utama Pengambilan Data JSON.....	85
Kode Sumber 4.6. Kode Untuk Pembuatan Array yang Berisi Stopword.....	85
Kode Sumber 4.7. Kode Fungsi Penghapusan Stopword di dalam Artikel	86
Kode Sumber 4.8. Kode Untuk Analisa Parameter Term Frequency	87
Kode Sumber 4.9. Kode Untuk Analisa Parameter Inverse Document Frequency	87
Kode Sumber 4.10. Kode Untuk Analisa Parameter Inverse Document Length.....	88
Kode Sumber 4.11. Kode Untuk Perhitungan Parameter TF, IDF, dan Norm	88
Kode Sumber 4.12. Kode Untuk Membership Function Term Frequency di dalam Dokumen	90
Kode Sumber 4.13. Kode Untuk Membership Function Term Frequency di dalam Query.....	92
Kode Sumber 4.14. Kode Untuk Membership Function Inverse Document Frequency	93
Kode Sumber 4.15. Kode Untuk Membership Function Inverse Document Length.....	95
Kode Sumber 4.16. Kode Untuk Membership Function Inverse Query Length	96
Kode Sumber 4.17. Kode Untuk Implementasi Fuzzy Rule	97
Kode Sumber 4.18. Kode Untuk Mengambil Artikel dan Tokenisasi	98
Kode Sumber 4.19. Kode Untuk Mengambil Query dan Tokenisasi	98

Kode Sumber 4.20. Kode Untuk Melakukan Pembobotan Dokumen Terhadap Query.....	99
Kode Sumber 4.21. Kode Untuk Pembuatan Objek Artikel	100
Kode Sumber 4.22. Kode Untuk Pengurutan Artikel Berdasarkan Nilai Hasil Pembobotan	100
Kode Sumber 4.23. Kode Untuk Menentukan Jumlah Artikel yang Ingin Dikembalikan	101
Kode Sumber 4.24. Kode Untuk Menentukan Skema Response dari Web Service.....	101
Kode Sumber 4.25. Kode Untuk Menentukan Atribut-Atribut Koneksi ke Web Service.....	102
Kode Sumber 4.26. Kode Untuk Menentukan Parameter Masukan Web Service.....	102
Kode Sumber 4.27. Kode Untuk Melakukan Koneksi ke Web Service	103
Kode Sumber 4.28. Kode Untuk Melakukan Pengambilan Elemen dari Response Web Service	103
Kode Sumber 4.29. Kode Implementasi Slide Drawer Navigasi	105
Kode Sumber 4.30. Kode Implementasi Halaman Utama	105
Kode Sumber 4.31. Kode Implementasi Halaman Pemilihan Kategori	106
Kode Sumber 4.32. Kode Implementasi Halaman Menampilkan Artikel Berdasar Kategori	107
Kode Sumber 4.33. Kode Implementasi Halaman Pencarian Artikel	108
Kode Sumber 4.34. Kode Implementasi Halaman Detail Artikel	109

BAB I

PENDAHULUAN

Bab ini membahas garis besar penyusunan tugas akhir yang meliputi latar belakang, tujuan pembuatan, rumusan dan batasan permasalahan, metodologi penyusunan tugas akhir, dan sistematika penulisan.

1.1. Latar Belakang

Dari hari ke hari, kebutuhan manusia terhadap perkembangan teknologi informasi semakin meningkat. Hal ini salah satunya ditunjukkan dengan meningkatnya tren penggunaan *smartphone* dalam kehidupan sehari-hari. Begitu pula dengan penggunaan berbagai macam *search engine* untuk mencari informasi. Pencarian informasi lebih dimudahkan dengan adanya *search engine* Google. Pengguna dapat menuliskan keyword pada *search engine* yang disediakan oleh Google, kemudian mesin pencari segera menampilkan halaman-halaman yang relevan. Halaman-halaman yang ditampilkan oleh *search engine* berasal dari beberapa website, terkadang pengguna tidak ingin direpotkan dengan mengunjungi halaman web tersebut satu-persatu untuk mendapatkan informasi. Sebuah informasi yang bersifat *categorical data* dapat dibuatkan sebuah aplikasi *search engine* tersendiri yang menghimpun informasi-informasi yang ada dari beberapa website untuk memudahkan pengguna dalam mencari informasi. *Categorical data* adalah informasi yang sudah dipisahkan berdasarkan kategorinya masing-masing [1]. Sebagai contoh adalah aplikasi Flipboard. Aplikasi ini adalah aplikasi penyedia artikel berita yang mengambil data-data artikel dari *social media* dan website-website lain [2].

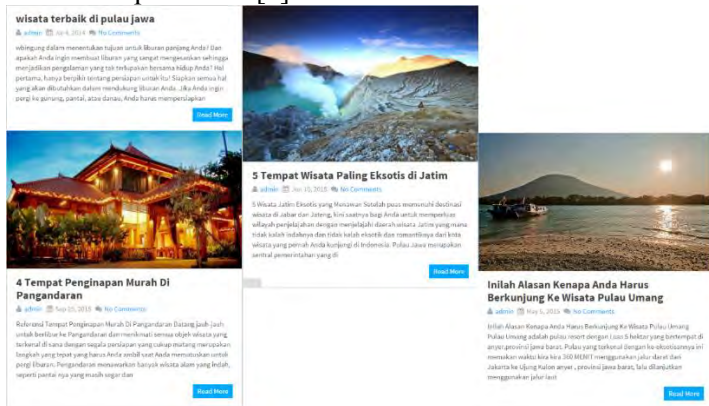
Selain informasi terhimpun, pengguna juga membutuhkan informasi yang relevan dari sebuah mesin pencari, Hasil pencarian harus diurutkan berdasarkan relevansinya agar memudahkan pengguna untuk mendapatkan informasi. Wisataindonesia.co.id

adalah salah satu penyedia artikel dengan kategori pariwisata. Website tersebut telah dilengkapi sebuah *search engine* untuk membantu mencari informasi yang ada didalamnya. Gambar 1.1 menunjukkan artikel-artikel tidak relevan yang muncul pada website tersebut jika dilakukan pencarian dengan keyword ‘pantai jawa timur’. Halaman pertama dari hasil pencarian menampilkan 4 hasil tidak relevan dari keseluruhan 12 artikel, artikel-artikel yang relevan ditunjukkan pada Gambar 1.2. Sebuah artikel dapat dikatakan tidak relevan jika artikel yang bersangkutan tidak memberikan informasi berkaitan dengan *query* yang dimasukkan oleh user [3]. Contoh pada Gambar 1.1 terdapat artikel tentang “penginapan murah di pangandaran” yang sangat tidak berkaitan dengan *query* “pantai jawa timur”. Contoh artikel lain yang dikembalikan adalah “inilah alasan kenapa anda harus berkunjung ke wisata pulau umang”, walaupun mengandung unsur “pantai” tetapi artikel tersebut tidak mengandung unsur “jawa timur”.

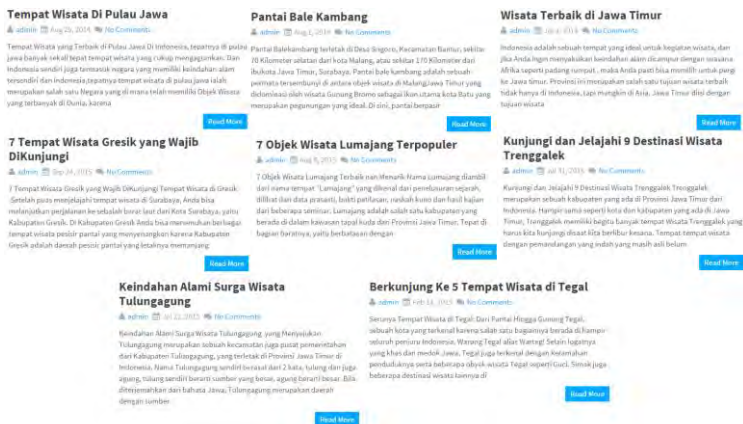
Oleh karena itu, dikembangkan suatu metode alternatif lain yang diharapkan dapat meningkatkan relevansi hasil pencarian. Tugas akhir ini akan mengimplementasikan *fuzzy logic based ranking function* kedalam suatu sistem pencarian artikel pariwisata berbasis mobile (Android). Sistem ini akan menghimpun artikel – artikel dari beberapa website yang menyediakan artikel terkait pariwisata. Data yang dihimpun dari website – website tersebut adalah judul artikel dan isinya.

Tujuan dari menerapkan *ranking function* ini adalah meningkatkan tingkat relevansi antara *input query* yang dimasukkan oleh user dengan dokumen artikel-artikel pariwisata yang ada. Diterapkannya *ranking function* yang berbasis fuzzy logic ini diharapkan meningkatkan tingkat *precision* dan *recall* dalam pengambilan artikel jika dibandingkan dengan *ranking function* yang menggunakan metode lain seperti TF – IDF (Okapi BM-25) yang sudah banyak digunakan [4]. Beberapa *preprocessing* seperti *stopword removal*, *word stemming*, perhitungan *tf*, perhitungan *idf*, perhitungan *jackard*, dan beberapa

preprocessing lainnya dapat dilakukan untuk meningkatkan relevansi hasil pencarian [5]



Gambar 1.1. Hasil search tidak relevan pada website wisataindonesia.co.id



Gambar 1.2. Hasil search relevan pada website wisataindonesia.co.id

1.2. Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir dapat dipaparkan sebagai berikut.

- a) Bagaimana mengambil data artikel dari sebuah website
- b) Bagaimana menentukan *membership function* dan *fuzzy rule* yang akan digunakan
- c) Bagaimana perancangan dan integrasi *web service* dengan perangkat Android
- d) Bagaimana menghitung tingkat relevansi dari *search engine*

1.3. Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir memiliki beberapa batasan, yakni sebagai berikut.

1. Implementasi dilakukan hanya untuk perangkat mobile (Android) dengan menggunakan *web service* lokal berbasis java
2. Implementasi *web service* menggunakan Spring Framework MVC
3. Data artikel merupakan artikel pariwisata berbahasa Indonesia
4. Data yang diolah hanya berupa data teks
5. Data yang diambil berasal dari
 - a. wisataindonesia.co.id
 - b. indonesiawisata.info
 - c. initempatwisata.com
 - d. anekatempatwisata.com

1.4. Tujuan

Tujuan dari pembuatan tugas akhir ini adalah mengimplementasikan *fuzzy based ranking function* kedalam suatu sistem pencarian artikel pariwisata berbasis perangkat mobile (Android) dengan harapan dapat meningkatkan relevansi antara hasil pencarian dan *input query* dari pengguna.

1.5. Metodologi

Tahap yang dilakukan untuk menyelesaikan tugas akhir ini adalah sebagai berikut:

1. Penyusunan Proposal Tugas Akhir

Di dalam proposal tugas akhir ini dijelaskan mengenai deskripsi pendahuluan dari tugas akhir yang akan dibuat. Pendahuluan berisi dari latar belakang alasan adanya usulan tugas akhir, rumusan masalah yang dibawa, batasan-batasan permasalahan, tujuan serta manfaat dari pembuatan tugas akhir ini. Sebagai bahan referensi pendukung dari usulan tugas akhir yang akan dibuat, maka dijelaskan pula tinjauan pustaka. Selain itu, terdapat metodologi sebagai langkah-langkah dalam pembuatan tugas akhir, dimulai dari penyusunan proposal hingga penyusunan buku tugas akhir dan sebagai haluan jadwal pelaksanaan tugas akhir dilampirkan pula jadwal kegiatan. Pada proposal ini, penulis mengajukan gagasan mengenai implementasi *fuzzy logic based ranking function* pada sistem pencarian artikel pariwisata berbasis perangkat *mobile* (Android).

2. Studi Literatur

Pada tahap ini dilakukan pencarian informasi dan studi literatur terkait beberapa teori pendukung serta pustaka pendukungnya yang antara lain algoritma perhitungan TF – idf, pendekatan *fuzzy*, penggunaan Import.io, serta beberapa library lain yang digunakan

3. Implementasi

Pada tahap ini dilakukan implementasi aplikasi pada *smartphone* dan dibangun dengan bahasa pemrograman Java berbasis Android, serta pembangunan *web service* berbasis Java Spring *framework* yang didalamnya

mengimplementasikan *fuzzy logic based ranking function*. Aplikasi ini dibangun dengan menggunakan *Integrated Development Environment* (IDE) Netbeans 8.1 untuk pembangunan *web service* dan Android Studio 1.4 untuk pembangunan aplikasi Android

4. Uji Coba dan Evaluasi

Pada tahap ini dilakukan uji coba aplikasi dan evaluasi terhadap implementasi metode pada aplikasi. Tahap uji coba dilakukan dengan menentukan beberapa *query* untuk uji coba dan membandingkan antara *input query* dengan hasil pencarian untuk mendapatkan nilai *precision* dan *recall*.

5. Penyusunan Buku Tugas Akhir

Tahap ini merupakan tahap dokumentasi dari tugas akhir. Buku Tugas Akhir berisi dasar teori, perancangan, implementasi dan hasil uji coba dan evaluasi dari aplikasi yang dibangun.

1.6. Sistematika Penulisan

Buku tugas akhir ini terdiri atas beberapa bab yang tersusun secara sistematis, yaitu sebagai berikut.

1. Bab I. Pendahuluan
Bab pendahuluan berisi penjelasan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan tugas akhir.
2. Bab II. Tinjauan Pustaka
Bab tinjauan pustakan berisi penjelasan mengenai dasar teori yang mendukung pengerjaan tugas akhir.
3. Bab III. Analisis dan Perancangan
Bab analisis dan perancangan berisi penjelasan mengenai analisis kebutuhan, perancangan sistem dan perangkat yang digunakan dalam pengerjaan tugas akhir serta urutan pelaksanaan proses.
4. Bab IV. Implementasi

Bab implementasi berisi pembangunan implementasi dengan algoritma *fuzzy based ranking function* pada sistem pencarian artikel pariwisata berbasis perangkat *mobile* sesuai dengan rumusan masalah dan batasan yang sudah dijelaskan pada bagian pendahuluan.

5. Bab V. Uji Coba dan Evaluasi

Bab uji coba dan evaluasi berisi pembahasan mengenai hasil dari uji coba yang dilakukan terhadap sistem pencarian artikel pariwisata yang didalamnya telah diimplementasikan *fuzzy logic based ranking function*.

6. Bab VI. Kesimpulan dan Saran

Bab kesimpulan dan saran berisi kesimpulan hasil penelitian. Selain itu, bagian ini berisi saran untuk pengerjaan lebih lanjut atau permasalahan yang dialami dalam proses pengerjaan tugas akhir.

[Halaman ini sengaja dikosongkan]

BAB II TINJAUAN PUSTAKA

Bab tinjauan pustaka berisi mengenai penjelasan teori yang berkaitan dengan implementasi perangkat lunak. Penjelasan tersebut bertujuan untuk memberikan gambaran mengenai sistem yang akan dibangun dan berguna sebagai pendukung dalam pengembangan perangkat lunak.

2.1. Pengambilan Data Menggunakan Import.io

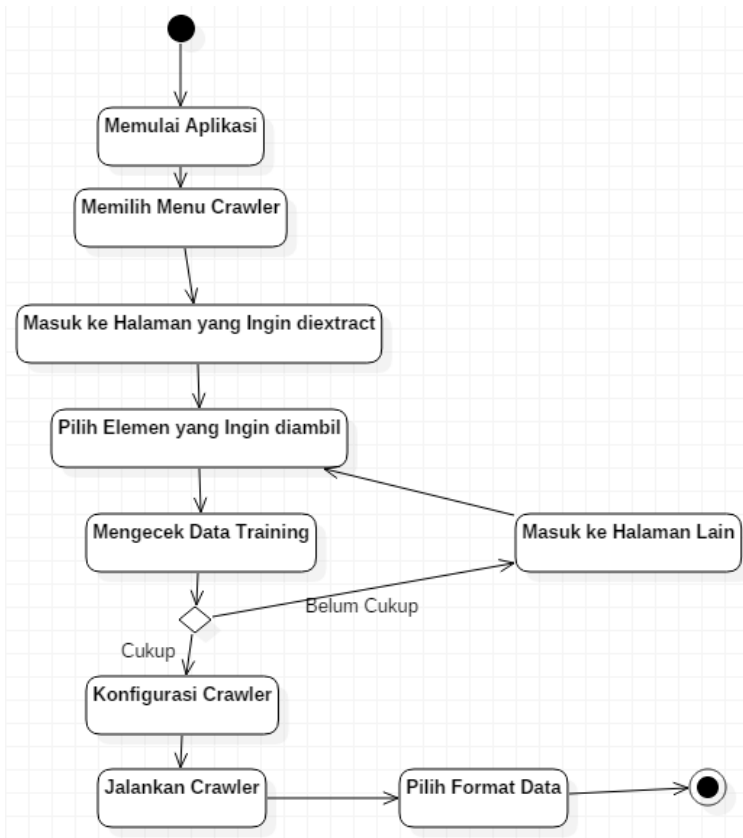
Import.io adalah sebuah *platform* yang memungkinkan pengguna untuk mengambil data dari web tanpa memerlukan pembuatan *custom code* [6]. Dengan menggunakan Import.io memungkinkan untuk melakukan *crawling* di sebuah website dan mendapatkan seluruh artikel yang ada didalam website tersebut. Pengguna dapat menentukan data apa saja yang ingin diambil dari sebuah halaman, seperti judul artikel, isi artikel, gambar, tanggal publikasi, penulis, dan berbagai informasi lainnya.

Penggunaan Import.io ini memberikan banyak kelebihan kepada penggunanya. Namun selain kelebihan yang ditawarkan, Import.io juga memiliki beberapa kekurangan. Masing-masing kelebihan dan kekurangannya antara lain sebagai berikut:

1. Kelebihan
 - a. Memudahkan pengambilan data pada website
 - b. Data disimpan dalam berbagai format (CSV, JSON, HTML)
 - c. Tersedia berbagai macam kustomisasi *crawler* (*starting point, depth of click*)
2. Kekurangan
 - a. Data yang diambil bersifat *raw*, sehingga terkadang tidak rapi
 - b. Tidak dapat mengambil *tag* HTML
 - c. Membutuhkan banyak data training untuk mendapatkan hasil *crawling* optimal

- d. Terdapat *data loss* jika crawling pada halaman yang memiliki format khusus

Pada tugas akhir ini, Import.io akan digunakan untuk melakukan *crawling* di website-website yang dijelaskan pada bagian batasan masalah. Data artikel yang didapatkan akan disimpan dalam file berformat JSON. Gambar 2.1 menunjukkan diagram aktivitas penggunaan aplikasi Import.io.



Gambar 2.1. Diagram Aktivitas Penggunaan Import.io

File JSON yang didapatkan dari hasil *crawling* menggunakan Import.io harus dirubah kedalam obek artikel agar dapat diproses ke tahap selanjutnya. Proses *parsing* file JSON dapat dilakukan dengan menggunakan library Java GSON. GSON yang juga dikenal dengan Google Gson adalah *library* Java *open source* yang digunakan untuk mengubah objek java menjadi JSON dan begitu juga sebaliknya [7]. Beberapa kelebihan GSON adalah:

1. Penggunaan yang mudah
2. Dapat menerima objek *custom*
3. Dapat melakukan *parsing* objek yang kompleks dengan mudah
4. File JSON yang dihasilkan bersifat padu dan mudah dibaca

Dalam tugas akhir ini objek artikel yang didapatkan dari hasil *parsing* menggunakan *library* GSON selanjutnya akan disimpan ke dalam *database* untuk diproses lebih lanjut.

2.2. Preprocessing Data

Data yang sudah didapatkan perlu dilakukan *preprocessing* agar dapat diproses pada tahap selanjutnya. Preprocessing yang dilakukan adalah penghapusan *stopword*, perhitungan TF – IDF, dan perhitungan *inverse document length*.

Stopword adalah kata – kata yang tidak penting atau tidak informatif. Data teks akan lebih baik jika dilakukan penghapusan *stopword* yang ada didalamnya agar bisa menghasilkan hasil pemrosesan yang lebih baik.

TF-IDF (*term frequency – inverse document frequency*) adalah suatu metode pembobotan yang sering digunakan dalam sistem temu kembali informasi dan *text mining*. Pembobotan ini adalah pengukuran secara statistik yang digunakan untuk mengevaluasi seberapa penting sebuah *term* terhadap dokumen yang ada dalam sebuah kumpulan dokumen [8].

Secara umum pembobotan TF-IDF terdiri dari dua buah komponen. Komponen pertama menghitung nilai yang sudah

dinormalisasikan dari frekuensi kemunculan suatu *term* yang muncul pada dokumen. Komponen kedua menghitung invers dari jumlah dokumen yang mengandung *term* bersangkutan.

Misalkan t_i adalah sebuah *term* i dalam *term collection* t dan d_j adalah sebuah dokumen j didalam *document collection* d . Fungsi f_{t_i, d_j} adalah fungsi untuk menghitung kemunculan suatu *term* t_i didalam dokumen d_j . Maka persamaan TF dapat dituliskan seperti pada persamaan 2.1:

$$TF(t_i, d_j) = \frac{f_{t_i, d_j}}{|\text{Jumlah term dalam dokumen } d_j|} \quad (2. 1)$$

Misalkan t_i adalah sebuah *term* i dalam *term collection* t , N adalah sekumpulan dokumen, dan DF_{t_i} adalah jumlah dokumen yang mengandung *term* i . Maka persamaan IDF dapat dituliskan seperti pada persamaan 2.2:

$$IDF(t_i) = \log \frac{|N|}{DF_{t_i}} \quad (2. 2)$$

Inverse document length atau yang biasa dikenal dengan *Field-length norm* sering digunakan untuk menghitung seberapa pentingnya suatu *term* didalam dokumen [9]. Semakin pendek sebuah dokumen, maka semakin tinggi nilai *norm*nya. Sebagai contoh, suatu *term* akan lebih berarti jika muncul pada bagian judul dari dokumen dibandingkan muncul pada bagian body dokumen.

Jika d_j adalah sebuah dokumen j didalam *document collection* d yang mengandung beberapa *term* maka persamaan *norm* dokumen dapat dituliskan pada Persamaan 2.3:

$$Norm(j) = \frac{1}{\sqrt{|\text{Jumlah term dalam dokumen } d_j|}} \quad (2. 3)$$

2.3. Pembobotan Dokumen TF – IDF

Dalam sistem temu kembali informasi TF-IDF (*term frequency-inverse document frequency*) merupakan sebuah nilai

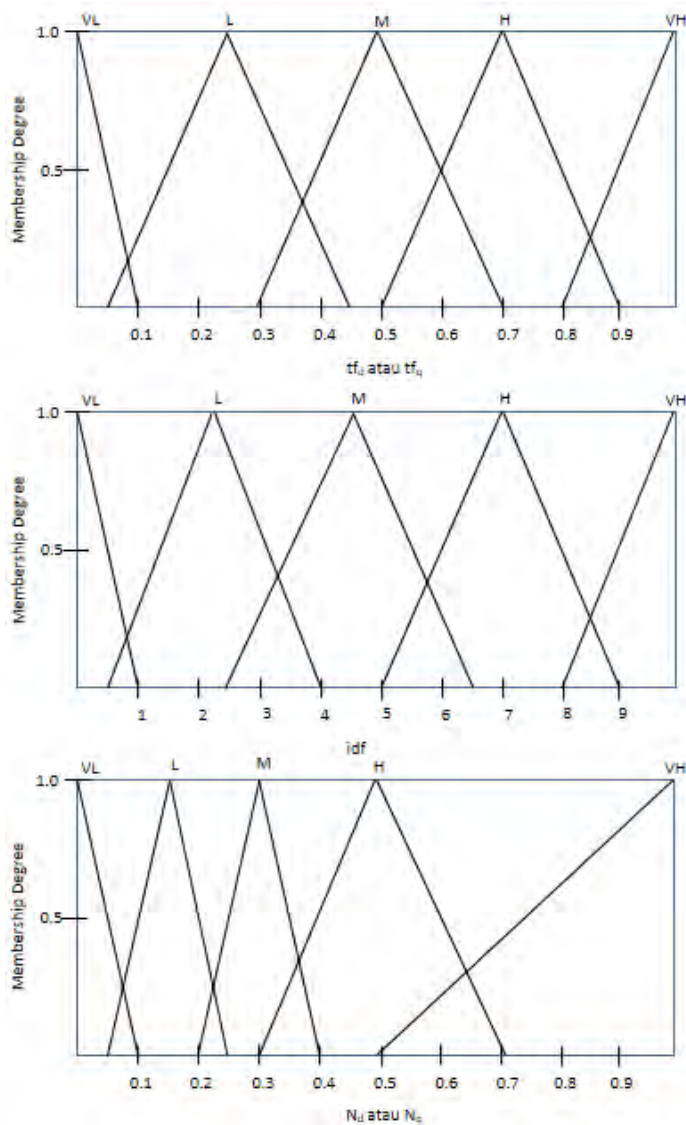
yang menunjukkan seberapa pentingnya sebuah *term* didalam kumpulan dokumen. Nilai TF-IDF meningkat secara proporsional seiring dengan meningkatnya jumlah kemunculan sebuah *term* didalam dokumen, peningkatan ini juga dipengaruhi dengan nilai IDF semakin sedikit jumlah dokumen yang mengandung *term* tertentu maka dapat diasumsikan bahwa *term* tersebut bersifat penting dan nilai IDF semakin meningkat.

Perhitungan TF – IDF dengan mengalikan nilai TF (*term frequency*) dengan IDF (*inverse document frequency*). Perhitungan nilai TF ditunjukkan pada Persamaan 2.1, sedangkan perhitungan IDF ditunjukkan pada Persaman 2.2.

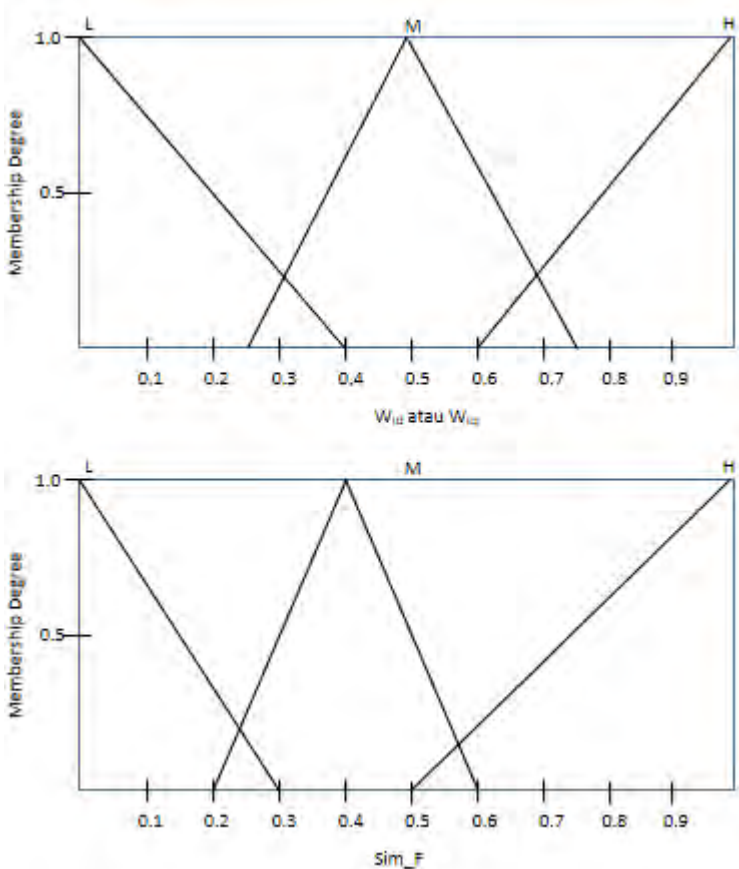
2.4. Pembobotan Dokumen Dengan Pendekatan Fuzzy

Logika *fuzzy* adalah peningkatan dari logika *Boolean* yang berhadapan dengan konsep kebenaran sebagian. Saat logika klasik menyatakan bahwa segala hal dapat diekspresikan dalam istilah biner (0 atau 1), logika *fuzzy* menggantikan kebenaran *Boolean* dengan tingkat kebenaran [10]

Logika *fuzzy* memiliki dua elemen yang penting dalam prosesnya. Pertama adalah *membership function* yang merupakan elemen untuk menentukan nilai keanggotaan sebuah parameter dalam kelas *fuzzy*. Kedua adalah *fuzzy rule* yang merupakan sekumpulan operasi yang digunakan untuk menentukan hasil dari logika *fuzzy*. Contoh *membership function* dari parameter-parameter yang dapat digunakan untuk pembobotan dokumen dengan pendekatan *fuzzy* ditunjukkan pada Gambar 2.2 dan Gambar 2.3.



Gambar 2.2. Contoh Membership Function Parameter untuk Pembobotan Dokumen (1)



Gambar 2.3. Contoh Membership Function Parameter untuk Pembobotan Dokumen (2)

Proses awal dari algoritma *fuzzy* adalah mengubah nilai *crisp* parameter kedalam nilai *fuzzy* sesuai dengan *membership function* yang ada. Pada Gambar 2.2 ditunjukkan contoh *membership function* untuk beberapa parameter. Pada contoh tersebut nilai TF (gambar teratas) dapat dirubah kedalam *fuzzy value* yang terdiri

dari 5 kelas (Very Low, Low, Medium, High, Very High). Begitu juga dengan parameter *idf* dan *inverse document length*.

Setelah mendapatkan *fuzzy value* dari setiap parameter, langkah selanjutnya adalah melakukan operasi dengan menggunakan *fuzzy rule*. *Fuzzy Rule* merupakan sekumpulan aturan yang disesuaikan dengan pengetahuan prosedural. *Rule* berfungsi untuk memproses informasi yang diberikan oleh parameter menjadi sebuah tindakan [11]. *Fuzzy rule* memiliki struktur yang terdiri dari satu atau beberapa *antecedent* pada bagian IF, dan sebuah *consequent* pada bagian THEN. Contoh fuzzy rule:

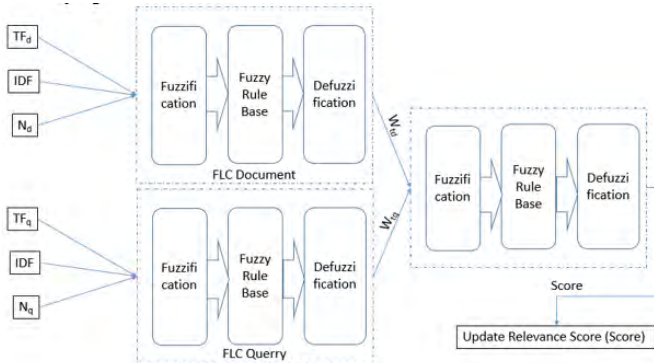
- IF TF Low AND IDF Low Then Score Low
- IF TF Medium AND IDF Medium Then Score Medium
- IF TF High AND IDF High Then Score High

Operasi dengan *fuzzy rule* akan menghasilkan beberapa *fuzzy value* dari masing-masing *rule*. Proses selanjutnya adalah melakukan *defuzzification* untuk mendapatkan hasil akhir *fuzzy*. Jika C_i adalah *fuzzy value* dari rule ke i dan Z_i adalah pemetaan *fuzzy value* dari rule i kedalam *crisp value*. Persamaan *defuzzification* dapat dituliskan pada persamaan 2.4:

$$Defuzzification(C_i) = \frac{\sum_{i=0}^n C_i Z_i}{\sum_{i=0}^n C_i} \quad (2.4)$$

Logika *fuzzy* juga dapat dilakukan untuk melakukan pembobotan dokumen terhadap suatu *term* yang diberikan. Hal ini dapat dilakukan dengan membuat suatu struktur *fuzzy* yang dikhususkan untuk pengolahan *query* dan dokumen [12]. Struktur *fuzzy* yang digunakan dalam tugas akhir ini merupakan *fuzzy* yang memiliki 2 tingkat. Pada tingkat pertama terdapat 2 buah *fuzzy logic controller* atau yang selanjutnya akan disebut FLC. FLC pertama digunakan untuk mengolah keterkaitan antara *term*

dengan dokumen, sedangkan FLC kedua digunakan untuk mengolah keterkaitan antara *term* dengan *query*. FLC pertama menerima 3 buah input, yaitu nilai *tf* sebuah *term* didalam dokumen, nilai *idf*, dan nilai *invers* dari panjang dokumen. FLC kedua juga menerima 3 buah input, yaitu nilai *tf* sebuah *term* didalam *query*. Tingkat kedua pada struktur *fuzzy* memiliki sebuah FLC yang digunakan untuk mengolah hasil keluaran dari dua FLC di tingkat pertama. Struktur *fuzzy* digambarkan pada Gambar 2.4 dan penjelasan parameter akan dijelaskan pada Tabel 2.1. Diagram yang menunjukkan hubungan tiap parameter pada struktur *fuzzy* ditunjukkan dengan menggunakan diagram permukaan pada Gambar 2.5.

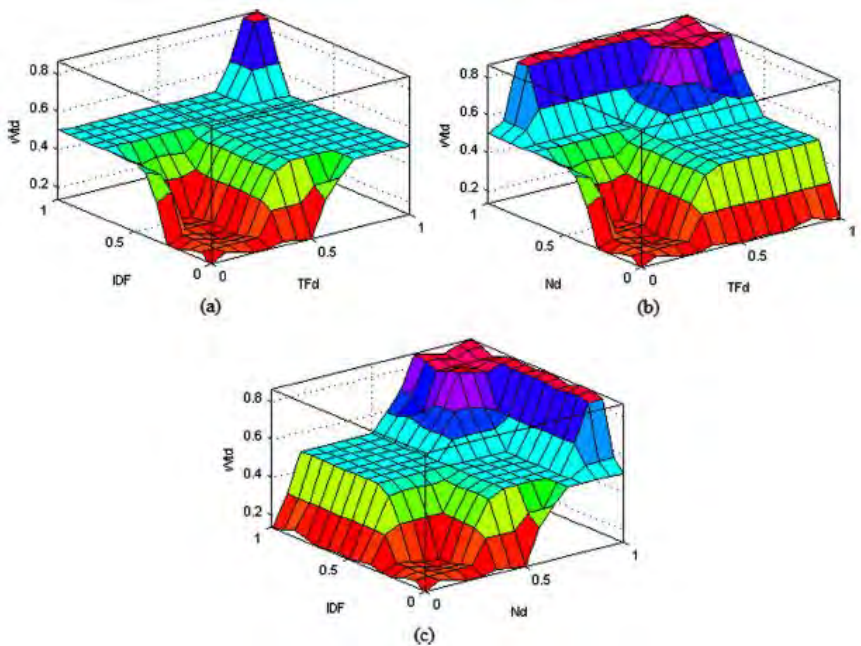


Gambar 2.4. Struktur Fuzzy

Tabel 2.1. Notasi didalam Struktur Fuzzy

No.	Nama Data	Keterangan	Jumlah Kelas Fuzzy
1.	TF_{td}	Frekuensi kemunculan suatu <i>term t</i> didalam dokumen <i>d</i>	5
2.	IDF_t	Nilai <i>invers</i> jumlah dokumen yang mengandung sebuah <i>term t</i>	5
3	N_d	Nilai <i>invers</i> panjang dokumen <i>d</i>	5

No.	Nama Data	Keterangan	Jumlah Kelas Fuzzy
4	TF_{tq}	Frekuensi kemunculan suatu <i>term t</i> didalam <i>query q</i>	5
5	N_q	Nilai <i>invers</i> panjang <i>query q</i>	5
6	W_{td}	Hasil keluaran dari FLC yang menghitung nilai relevansi <i>term t</i> terhadap dokumen <i>d</i>	3
7	W_{tq}	Hasil keluaran dari FLC yang menghitung nilai relevansi <i>term t</i> terhadap <i>query q</i>	3



Gambar 2.5. Hubungan Antar Parameter Dalam Fuzzy [12]

Contoh perhitungan pembobotan *fuzzy* dari sebuah *term* terhadap sebuah dokumen dengan menggunakan *membership function* pada Gambar 4 dan Gambar 5 ditunjukkan pada Tabel 2.2.

Tabel 2.2. Contoh Perhitungan Pembobotan Fuzzy

Artikel (d)	Jawa Timur, provinsi ini seolah memiliki magnet bagi para wisatawan domestik maupun mancanegara. Keistimewaan pantai ini adalah pasir putihnya nan lembut, air laut jernih berwarna toska dan juga air terjun setinggi 8 meter.	
Query (q)	Pantai	
Parameter	$TF_{td} = 32/625 = 0.051$ (Persamaan 2.1) $TF_{tq} = 1/1 = 1$ (Persamaan 2.1) $IDF_t = 0.667$ (Persamaan 2.2) $N_d = 1/\sqrt{625} = 0.040$ (Persamaan 2.3) $N_q = 1/\sqrt{1} = 1$ (Persamaan 2.3)	
Fuzzification TF_{td}	Very Low	0.49
	Low	0.01
	Medium	0
	High	0
	Very High	0
Fuzzification TF_{tq}	Very Low	0
	Low	0
	Medium	0
	High	0
	Very High	1
Fuzzification IDF_t	Very Low	0.33
	Low	0.11
	Medium	0
	High	0
	Very High	0
	Very Low	0.6

Fuzzification N_d	Low	0
	Medium	0
	High	0
	Very High	0
Fuzzification N_q	Very Low	0
	Low	0
	Medium	0
	High	0
	Very High	1
Rule Level 1	<ul style="list-style-type: none"> • IF TF Very Low AND IDF Very Low AND N Very Low THEN W Low • IF TF Low AND IDF Low AND N Very Low THEN W Low • IF TF Very High AND IDF Very Low AND N Very High THEN W Medium • IF TF Very High AND IDF Low AND N Very High THEN W Medium 	
FLC Document	Rule 1: $\text{Min}(0.49, 0.33, 0.6) = 0.33$ Rule 2: $\text{Min}(0.01, 0.11, 0.6) = 0.01$ Rule 3: $\text{Min}(0, 0.33, 0) = 0$ Rule 4: $\text{Min}(0, 0.11, 0) = 0$	
Crisp Value	Rule 1: $0.4 - (0.33 * 0.4) = 0.27$ Rule 2: $0.4 - (0.01 * 0.4) = 0.40$	
Defuzzification	Output = $(0.33 * 0.27 + 0.01 * 0.40) / (0.33 + 0.01) = 0.27$ (Persamaan 2.4)	
FLC Query	Rule 1: $\text{Min}(0, 0.33, 0) = 0$ Rule 2: $\text{Min}(0, 0.11, 0) = 0$ Rule 3: $\text{Min}(1, 0.33, 1) = 0.33$ Rule 4: $\text{Min}(1, 0.11, 1) = 0.11$	
Crisp Value	Rule 3: $0.25 + (0.25 * 0.33) = 0.33$	

	Rule 4: $0.25 + (0.25 * 0.11) = 0.28$	
Defuzification	Output = $(0.33 * 0.33 + 0.11 * 0.28) / (0.33 + 0.28) = 0.23$	
Fuzzification W_{td}	Low	0.32
	Medium	0.08
	High	0
Fuzzification W_{tq}	Low	0.42
	Medium	0
	High	0
Rule Level 2	<ul style="list-style-type: none"> IF W_{td} Low AND W_{tq} Low THEN F Low IF W_{td} Low AND W_{tq} Medium THEN F Medium 	
FLC Output	Rule 1: $\text{Min}(0.32, 0.42) = 0.32$ Rule 2: $\text{Min}(0.08, 0.42) = 0.08$	
Crisp Value	Rule 1: $0.3 - (0.3 * 0.32) = 0.20$ Rule 2: $0.2 + (0.2 * 0.08) = 0.22$	
Defuzification	Output = $(0.32 * 0.20 + 0.08 * 0.22) / (0.32 + 0.08) = 0.204$ (Persamaan 2.4)	
Bobot Akhir	0.204	

2.5. Komunikasi Web Service dengan Android

Web Service adalah sebuah atau beberapa aplikasi yang dapat diakses lewat internet dan menggunakan standar sistem komunikasi XML [13]. Komunikasi ke *web service* dapat dilakukan oleh *client* dengan cara mengirimkan pesan XML kepada *server* yang menyediakan *web service*, kemudian setelah beberapa waktu *server* akan mengembalikan *response* yang berbentuk XML kepada *client*. Seluruh komunikasi didalam *web service* menggunakan format XML sehingga tidak terikat pada sebuah sistem operasi atau Bahasa pemrograman.

Web Service memiliki beberapa komponen, yaitu:

- SOAP (*Simple Object Access Protocol*), yang digunakan sebagai standar untuk mengirim dan menerima pesan.
- XML (*Extensible Markup Language*), yang digunakan sebagai format data.
- WSDL (*Web Services Description Language*), yang mendeskripsikan ketersediaan dari sebuah *web service*.

Sebuah *web service* juga dapat terintegrasi dengan database. Netbeans 8.1 menyediakan berbagai macam *driver* untuk koneksi ke *server database* Java DB, MySQL, Oracle, PostgreSQL [14]. Dengan *driver* yang disediakan, koneksi antara *web service* dengan *database* dapat dengan mudah dilakukan. Pada tugas akhir ini pengembangan *web service* akan digunakan IDE Netbeans 8.1 dan *database* MySQL.

Dalam tugas akhir ini *web service* akan melakukan komunikasi dengan aplikasi di perangkat Android. Segala perhitungan dilakukan pada sisi *web service* sedangkan aplikasi di perangkat Android hanya berfungsi untuk mengirim data ke *web service* dan menampilkan hasil pengolahan data.

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti *smartphone* dan telepon *tablet*. Antarmuka pengguna Android umumnya berupa manipulasi langsung menggunakan gerakan sentuh yang bersifat *real-time* seperti menggeser, mengetuk, *pinch*, dan manipulasi objek di layar [15]. Arsitektur Android terdiri dari layer aplikasi, yang merupakan tempat berjalannya aplikasi seperti kontak, *phone*, dan *browser*. Setelah layer *application*, terdapat layer *application framework*, *libraries*, *Android runtime*, dan *kernel* Linux.

Pengembangan aplikasi Android dapat dilakukan dengan menggunakan IDE (*Integrated Development Environment*) Android Studio yang disediakan oleh Google [16]. IDE ini dikembangkan dari IntelliJ IDEA milik JetBrains. Beberapa fitur yang disediakan oleh Android Studio adalah:

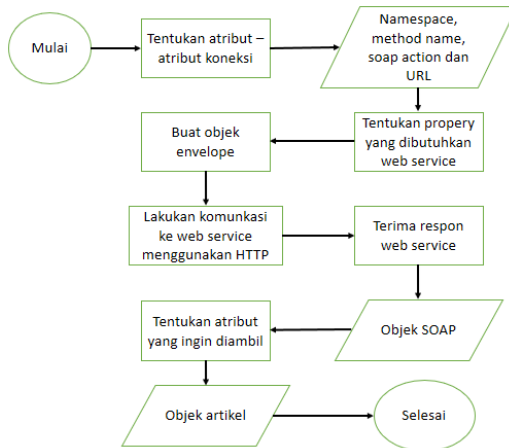
- Mendukung *build* berbasis *gradle*
- Refactoring dan *quick fixes* yang disesuaikan untuk Android
- *Signing* Aplikasi
- *Template based wizard* untuk memudahkan membangun UI Android

Aplikasi Android yang dibangun memerlukan untuk berkomunikasi dengan sebuah *web service* yang berbasis SOAP. Untuk memungkinkan hal ini maka dibutuhkan bantuan *library* KSOAP2 yang digunakan untuk berkomunikasi dengan *web service* berbasis SOAP. KSOAP2 adalah *library* Android yang digunakan untuk komunikasi perangkat Android dengan *web service* SOAP [17]. KSOAP2 terdiri dari XML *parser*, *serializer* dan *deserializer*, serta berfungsi sebagai *transport layer*. KSOAP2 melakukan *parsing* XML menggunakan *parser* kXML.

KSOAP2 sebagai *serializer* dan *deserializer* dapat melakukan *mapping* sebuah objek java kedalam format XML dan begitu juga sebaliknya. Kemudian *transport layer* pada KSOAP2 menyediakan mekanisme komunikasi antara *client* dan *server* pada protokol SOAP. Mekanisme yang disediakan dalam *library* ini adalah:

- *HttpTransport* untuk *platform* J2ME
- *HttpTransportSE* untuk *platform* J2SE

Dalam penggunaannya, *library* KSOAP2 diharuskan menjalankan melalui *thread* baru karena dalam pemrograman Android koneksi ke *network* tidak boleh dilakukan di *thread* utama. Diagram alir penggunaan *library* KSOAP2 ditunjukkan pada Gambar 2.6.



Gambar 2.6. Diagram Alir Penggunaan KSOAP2

BAB III

ANALISIS DAN PERANCANGAN

Pada Bab 3 ini akan dijelaskan mengenai analisis dan perancangan perangkat lunak untuk mencapai tujuan dari Tugas Akhir. Perancangan ini meliputi perancangan data, perancangan proses, dan perancangan antar muka, serta juga akan dijelaskan tentang analisis implementasi metode secara umum pada sistem.

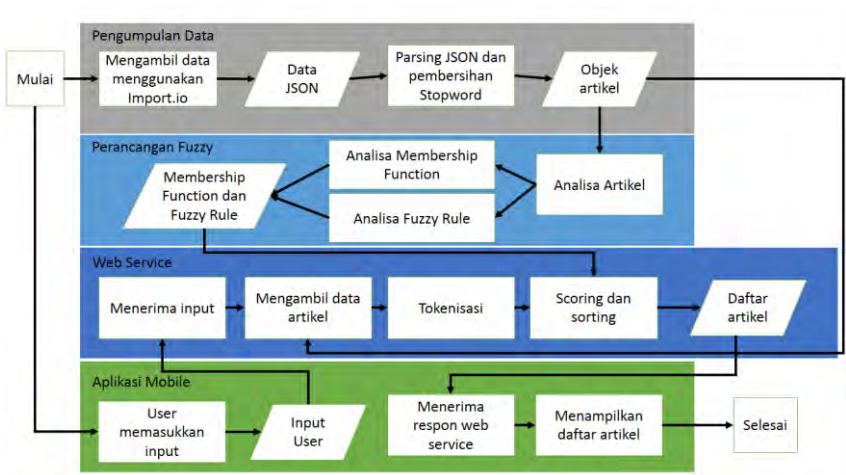
3.1. Analisis Implementasi Secara Umum

Pada tugas akhir ini akan dibangun sebuah sistem untuk melakukan pencarian artikel pariwisata dalam sebuah perangkat *mobile* menggunakan *ranking function* yang berbasis *fuzzy logic*. Proses-proses yang terlibat di dalam implementasi sistem ini meliputi tahap pengumpulan data, penghapusan *stopword*, perancangan algoritma *fuzzy*, pembuatan *web service*, pembuatan serta integrasi aplikasi perangkat *mobile* dengan *web service*. Tahap pengumpulan data adalah tahap pertama yang dilakukan sebelum dilakukan proses implementasi sistem. Data yang diambil adalah data-data artikel bertopik pariwisata, pengambilan data dilakukan dengan menggunakan *tools crawler* Import.io. Data yang diambil berbentuk file JSON yang didalamnya berisi seluruh detail dari artikel yang ada di *website*. File JSON yang didapatkan selanjutnya akan diparsing menjadi objek artikel yang berformat judul dan isi dengan menggunakan *library* GSON. Objek artikel yang didapatkan selanjutnya akan dibersihkan dari *stopword* yang dimana daftar *stopword* tersimpan dalam sebuah file txt. Setelah artikel sudah bersih dari *stopword*, kemudian artikel akan dimasukkan kedalam *database*. Setelah data-data artikel telah didapatkan, tahap selanjutnya adalah merancang algoritma *fuzzy* yang disesuaikan dengan data artikel. Tahap perancangan algoritma *fuzzy* meliputi perancangan *membership function* untuk setiap parameter dan perancangan *rule-rule fuzzy* apa saja yang akan digunakan. Tahap selanjutnya adalah membuat *web service* menggunakan algoritma *fuzzy* yang sudah dirancang pada tahap

sebelumnya. *Web service* yang telah dirancang memiliki alur sebagai berikut:

- Menerima input dari *user*
- Mengambil data artikel dari *database*
- Melakukan tokenisasi *query user* dan data artikel
- Melakukan perhitungan *score* untuk tiap artikel
- Melakukan pengurutan berdasarkan *score* dari setiap artikel
- Mengembalikan daftar artikel ke *user*

Tahap selanjutnya setelah melakukan pembuatan *web service* adalah membuat aplikasi di perangkat *mobile* dan mengintegrasikan dengan *web service*. Aplikasi *mobile* yang akan digunakan nantinya berfungsi untuk mengirimkan *request* dan menerima respon dari *web service*. Komunikasi antara perangkat *mobile* dengan *web service* dilakukan dengan bantuan *library* KSOAP2. Diagram alir dari keseluruhan proses implementasi sistem dan alur program digambarkan pada Gambar 3.1.



Gambar 3.1. Diagram Alir Implementasi Program

3.2. Perancangan Data

Pada subbab ini akan dibahas mengenai perancangan data yang merupakan bagian penting karena data sebagai objek yang akan diolah oleh perangkat lunak dalam tugas akhir ini dan menghasilkan sebuah informasi. Data yang akan digunakan pada sistem ini adalah data masukan (*input*), data proses, dan data keluaran (*output*) yang memberikan hasil pengolahan sistem ini untuk pengguna.

3.2.1. Data Masukan

Data masukan merupakan data awal yang akan diproses oleh sistem untuk melakukan pencarian. Ada 2 data masukan berbentuk teks yang akan diproses, pertama adalah data *query* pengguna yang dikirim dari perangkat *mobile*. Data kedua adalah data–data artikel yang ada didalam *database*. Data artikel yang terdapat dalam database berjumlah 534 artikel, ditambah dengan input dari pengguna maka total terdapat 535 data masukan yang akan diolah.

3.2.2. Data Proses

Data proses adalah data yang digunakan selama proses berjalannya sistem yang merupakan hasil pengolahan dari data masukan untuk diproses kembali menjadi data keluaran di tahap selanjutnya. Data proses yang digunakan di dalam proses ini ditunjukkan oleh Tabel 3.1.

Tabel 3.1. Data Proses

No.	Nama Data	Keterangan
1.	documentToken	Hasil tokenisasi isi artikel yang berisi token–token dari isi artikel
2.	titleToken	Hasil tokenisasi judul artikel yang berisi token–token dari judul artikel

No.	Nama Data	Keterangan
3	queryToken	Hasil tokenisasi <i>input query</i> dari pengguna yang berisi token–token dari <i>query</i>
4	tfDoc	Nilai <i>tf (term frequency)</i> dari sebuah term pada dokumen tertentu
5	tfQue	Nilai <i>tf (term frequency)</i> dari sebuah term pada dokumen tertentu
6	idf	Nilai <i>idf (inverse document frequency)</i> dari sebuah term pada sekumpulan dokumen
7	nDoc	Nilai <i>norm</i> atau <i>inverse length</i> dari sebuah dokumen
8	nQue	Nilai <i>norm</i> atau <i>inverse length</i> dari sebuah <i>query</i>
9	weightDoc	Hasil dari operasi <i>fuzzy logic controller</i> yang mengolah dokumen
10	weightQue	Hasil dari operasi <i>fuzzy logic controller</i> yang mengolah <i>query</i>
11	score	Nilai akhir pembobotan sebuah term terhadap dokumen
12	tfDocM	Hasil pemetaan <i>crisp value</i> tfDoc kedalam <i>fuzzy value</i> kedalam kelas–kelas berdasarkan <i>membership function</i>
13	tfQueM	Hasil pemetaan <i>crisp value</i> tfQue kedalam <i>fuzzy value</i> kedalam kelas–kelas berdasarkan <i>membership function</i>
14	idfM	Hasil pemetaan <i>crisp value</i> idf kedalam <i>fuzzy value</i> kedalam kelas–kelas berdasarkan <i>membership function</i>
15	nDocM	Hasil pemetaan <i>crisp value</i> nDoc kedalam <i>fuzzy value</i> kedalam kelas–

No.	Nama Data	Keterangan
		kelas berdasarkan <i>membership function</i>
16	nQueM	Hasil pemetaan <i>crisp value</i> nQue kedalam <i>fuzzy value</i> kedalam kelas–kelas berdasarkan <i>membership function</i>
17	weightDocM	Hasil pemetaan <i>crisp value</i> weightDoc kedalam <i>fuzzy value</i> kedalam kelas–kelas berdasarkan <i>membership function</i>
18	WeightQueM	Hasil pemetaan <i>crisp value</i> weightQue kedalam <i>fuzzy value</i> kedalam kelas–kelas berdasarkan <i>membership function</i>
19	fuzzyDocRule	Hasil operasi <i>fuzzy rule</i> pada <i>fuzzy logic controller</i> dokumen
20	fuzzyQueRule	Hasil operasi <i>fuzzy rule</i> pada <i>fuzzy logic controller query</i>
21	fuzzyScoreRule	Hasil operasi <i>fuzzy rule</i> pada <i>fuzzy logic controller</i> terakhir
22	docZValue	Hasil pemetaan <i>fuzzy value</i> ke <i>crisp value</i> pada <i>fuzzy logic controller</i> dokumen
23	queZValue	Hasil pemetaan <i>fuzzy value</i> ke <i>crisp value</i> pada <i>fuzzy logic controller query</i>
24	scoreZValue	Hasil pemetaan <i>fuzzy value</i> ke <i>crisp value</i> pada <i>fuzzy logic controller</i> terakhir

3.2.3. Data Keluaran

Data keluaran dari sistem ini adalah berupa sekumpulan objek artikel yang memiliki atribut seperti ditunjukkan pada Tabel 3.2.

Tabel 3.2. Data Keluaran

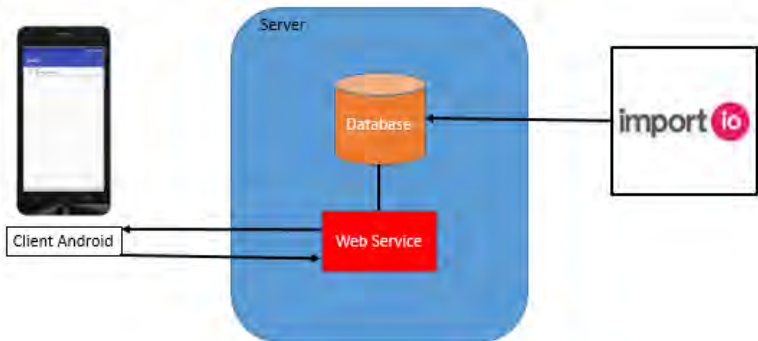
No.	Nama Data	Keterangan
1.	articleIndex	Atribut yang menyimpan nilai id artikel didalam <i>database</i>
2.	articleScore	Atribut yang menyimpan nilai hasil pembobotan antara artikel dengan <i>term</i>
3	articleTitle	Atribut yang menyimpan judul dari artikel
4	articleContent	Atribut yang menyimpan isi dari artikel

Data keluaran yang berbentuk objek selanjutnya akan ditampilkan kedalam *listview* yang ada di perangkat *mobile*. Tampilan data keluaran pada perangkat *mobile* ditunjukkan pada Gambar 3.2.

**Gambar 3.2. Data Keluaran di Perangkat Mobile**

3.3. Perancangan Proses

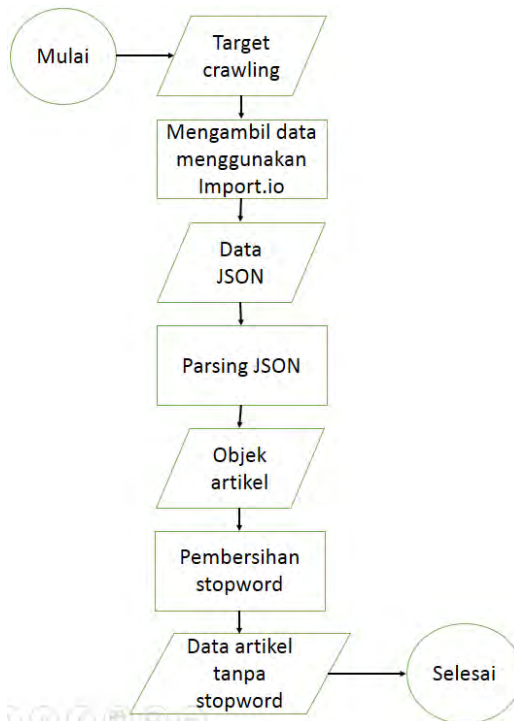
Pada subbab ini akan dibahas mengenai perancangan proses yang dilakukan untuk memberikan gambaran secara rinci pada setiap alur implementasi metode pada sistem pencarian artikel pariwisata ini. Alur tersebut nantinya akan digunakan dalam tahap implementasi. Arsitektur keseluruhan dari sistem yang diimplementasikan ditunjukkan pada Gambar 3.3.



Gambar 3.3. Arsitektur Sistem Pencarian Artikel

3.3.1. Tahap Pengumpulan dan Persiapan Data

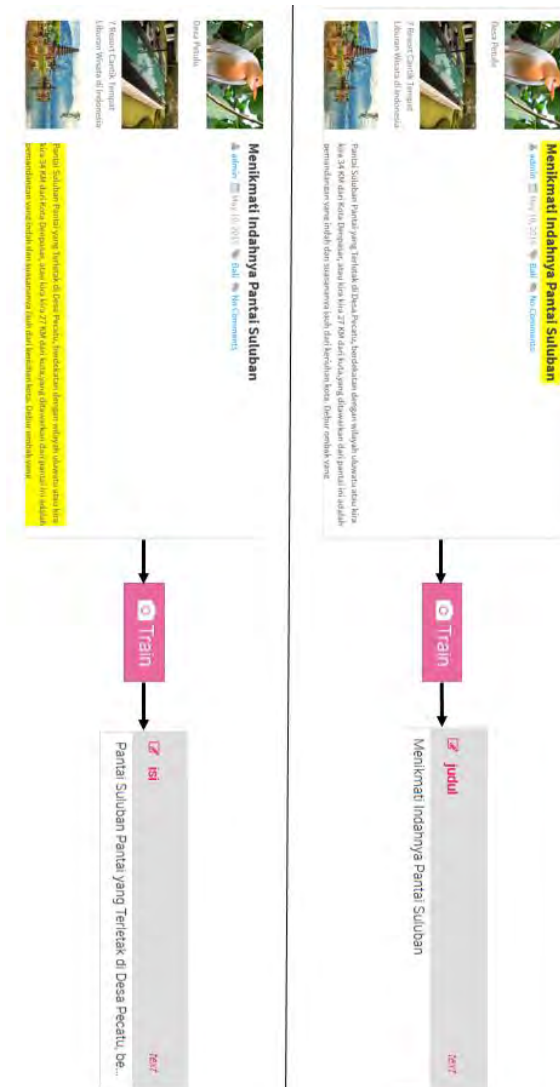
Di dalam tahap pengumpulan data ini terdapat beberapa proses antara lain pengambilan data artikel menggunakan Import.io, *parsing* data JSON kedalam objek artikel, dan pembersihan *stopword* yang ada didalam artikel. Diagram alir mengenai tahap ini dapat dilihat pada Gambar 3.4.



Gambar 3.4. Diagram Alir Tahap Pengumpulan Data

3.3.1.1. Pengambilan Data Artikel Menggunakan Import.io

Pengambilan data artikel menggunakan Import.io adalah proses pertama di dalam tahap pengumpulan data. Data yang diambil berupa data teks masing-masing merupakan judul dan isi dari artikel. Diagram alir dari proses pengambilan data menggunakan Import.io ditunjukkan pada Gambar 3, pada gambar tersebut dapat dilihat terdapat proses pengecekan data training. Data training ini akan *digenerate* secara otomatis oleh Import.io berdasarkan elemen apa saja yang ingin diambil oleh pengguna. Contoh pemilihan elemen pada Import.io ditunjukkan pada Gambar 3.5.

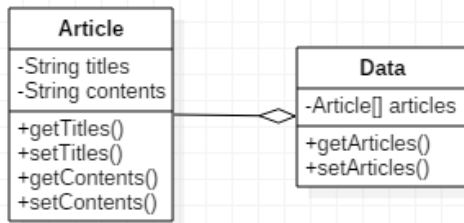


Gambar 3.5. Proses Training Pada Import.io

3.3.1.2. Parsing Data JSON Kedalam Objek Artikel

Data artikel yang diambil menggunakan Import.io akan disimpan kedalam format JSON. Contoh hasil pengambilan data ditunjukkan pada Gambar 13. Dalam gambar tersebut dapat dilihat terdapat banyak atribut untuk setiap artikel, tetapi untuk selanjutnya hanya atribut 'judul' dan 'isi' yang akan diproses. Agar data artikel dapat diproses pada tahap selanjutnya, maka objek artikel harus didapatkan dari file JSON tersebut. Pada tahap ini akan dilakukan proses parsing data JSON kedalam objek artikel menggunakan library GSON.

Untuk melakukan *parsing* file JSON kedalam objek artikel dengan menggunakan GSON, langkah pertama yang harus dilakukan adalah Mengamati struktur file JSON yang didapatkan dan posisi data yang ingin diambil. Gambar 3.7 menunjukkan seluruh data judul dan isi dari artikel berada di dalam *tag* data. Maka struktur dari artikel dapat ditunjukkan sebagai diagram kelas pada Gambar 3.6.



Gambar 3.6. Struktur Objek Artikel

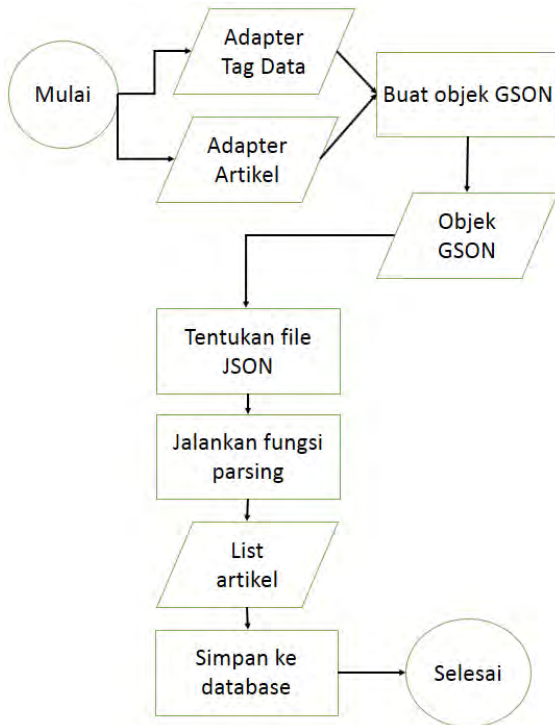
```

"data": [
  {
    "judul": [
      "Info Tempat Wisata Jambi"
    ],
    "isi": [
      "Tempat Wisata Jambi - Jambi merupakan kota yang memiliki ibu kc
    ],
    "_resultNumber": 1,
    "_widgetName": "Crawler_WisataIndonesia",
    "_source": [
      "5c4cb051-ba26-4cb6-9b3b-cf5220200077"
    ],
    "_connectorVersionGuid": "0a5902c4-d94c-4ec3-9137-c337ba4b6580",
    "_pageUrl": "http://wisataindonesia.co.id/tempat-wisata-jambi",
    "_outputTypes": {
      "judul": "STRING",
      "isi": "STRING"
    },
    "_input": {
      "
    },
    "_num": 1
  },
],

```

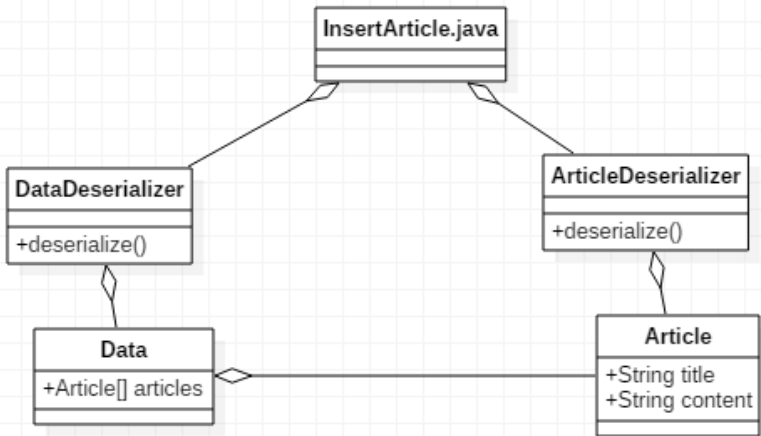
Gambar 3.7. Hasil Crawling Menggunakan Import.io

Setelah menentukan struktur file JSON dan letak data yang ingin diambil, proses selanjutnya adalah membuat *custom adapter* yang digunakan untuk melakukan pengambilan seluruh data artikel. Karena posisi seluruh data artikel ada di dalam *tag* data, maka dibutuhkan dua *custom adapter* yang pertama untuk mengambil seluruh artikel yang ada di dalam *tag* data dan yang kedua untuk mengambil data judul dan isi artikel. Setelah *custom adapter* selesai dibuat, proses selanjutnya adalah menentukan file JSON yang akan diparsing. Proses ini akan dilakukan sebanyak 4 kali karena dalam tugas akhir ini dilakukan *crawling* ke 4 website. Setelah memilih file JSON maka proses *parsing* dapat dilakukan. Diagram alir penggunaan library GSON ditunjukkan pada Gambar 3.8.



Gambar 3.8. Diagram Alir Penggunaan Library GSON

Program yang digunakan untuk melakukan parsing data JSON kedalam objek artikel memiliki struktur kelas yang dapat digambarkan seperti pada Gambar 3.9.



Gambar 3.9. Diagram Kelas Program Parsing Data JSON ke Objek Artikel

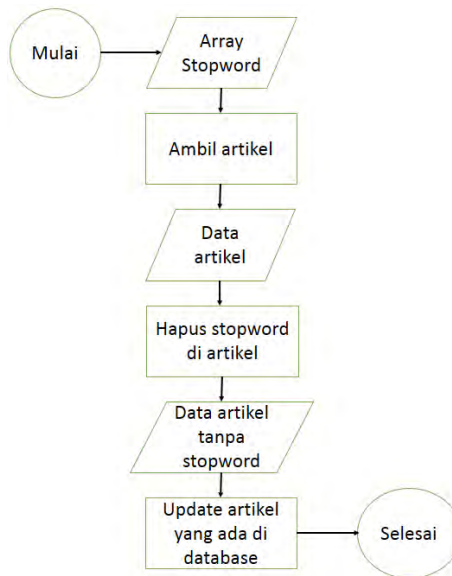
Kelas-kelas yang ditunjukkan pada Gambar 3.9 dapat dijelaskan sebagai berikut:

- **InsertArticle** merupakan kelas yang mengimplementasikan diagram alir pada Gambar 3.8.
- **DataDeserIALIZER** merupakan kelas yang digunakan untuk mengambil seluruh artikel yang berada dalam *tag* 'data' seperti yang ditunjukkan pada Gambar 3.7.
- **Data** merupakan kelas yang merepresentasikan objek hasil pengambilan dari *tag* 'data' yang merupakan sekumpulan artikel.
- **ArticleDeserIALIZER** merupakan kelas yang digunakan untuk mengambil elemen judul dan isi artikel dari data JSON.
- **Article** merupakan kelas yang merepresentasikan objek artikel yang memiliki atribut judul dan isi.

3.3.1.3. Pembersihan Stopword

Data artikel sebagai data keluaran dari proses *parsing* menggunakan *library* GSON saja masih belum cukup baik untuk

diproses pada tahap selanjutnya. Hal tersebut dikarenakan jumlah kata didalam artikel masih relatif terlalu banyak untuk diproses. Terutama jumlah kata yang ada di bagian isi dari artikel. Jumlah kata yang banyak itu dikarenakan data artikel tersebut masih mengandung banyak *stopword*, baik itu pada bagian isi artikel maupun pada bagian judul artikel. Untuk mengatasi hal tersebut, pada tugas akhir ini akan diterapkan *stopword removal* untuk meningkatkan hasil pembobotan artikel.



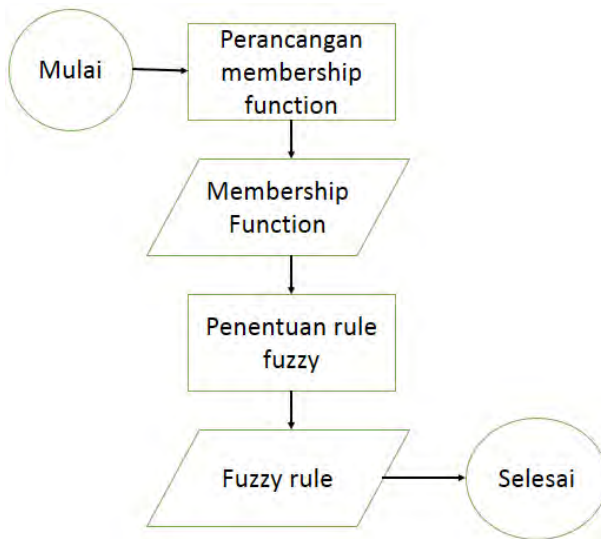
Gambar 3.10. Diagram Alir Proses Stopword Removal

Pada alur proses penerapan *stopword removal* pertama kata - kata yang termasuk *stopword* akan disimpan dalam sebuah *array*. Langkah berikutnya data artikel yang ada di *database* diambil dan dilakukan iterasi setiap artikelnya dan menghapus kata-kata yang termasuk didalam *array stopwords*. Kemudian artikel dalam *database* akan diupdate dengan data artikel yang

sudah bersih dari *stopword*. Diagram alir proses *stopword removal* ditunjukkan pada Gambar 3.10.

3.3.2. Tahap Perancangan Pembobotan Dokumen dengan Pendekatan Fuzzy

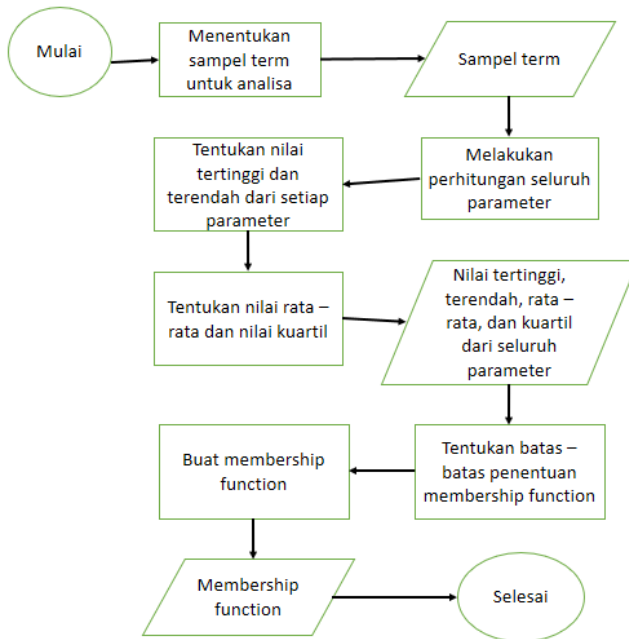
Di dalam tahap perancangan pembobotan dengan pendekatan *fuzzy* ini dilakukan proses perancangan struktur *fuzzy* yang disesuaikan dengan dataset yang dihasilkan pada tahap pertama yakni tahap pengambilan data menggunakan Import.io. Struktur *fuzzy* yang akan diimplementasikan ditunjukkan pada Gambar 6. Pada tahap ini terdapat dua bahasan utama, yaitu perancangan *membership function*, dan penentuan *fuzzy rule*. Diagram alir dari tahap perancangan pembobotan *fuzzy* ini dapat dilihat pada Gambar 3.11.



Gambar 3.11. Diagram Alir Tahap Perancangan Fuzzy

3.3.2.1. Perancangan Membership Function

Di dalam proses ini dilakukan analisa setiap parameter yang digunakan dalam algoritma *fuzzy*. Analisa ini penting dilakukan karena pemetaan dari *crisp value* kedalam *fuzzy value* dapat sesuai dengan dataset artikel yang didapatkan. Diagram alir proses perancangan *membership function* ditunjukkan pada Gambar 3.12.



Gambar 3.12. Diagram Alir Proses Perancangan Membership Function

3.3.2.2.1. Hasil Analisa Parameter

Di bagian ini akan dilakukan analisa parameter *fuzzy*. Beberapa parameter seperti *term frequency* dan *inverse document frequency* membutuhkan sampel *query* untuk melakukan analisa. Sampel *query* yang dipakai hanya *query* yang berisi satu buah kata dan untuk meningkatkan hasil analisa maka digunakan kata-kata

yang berkaitan dengan pariwisata dan memiliki nilai rata-rata *term frequency* dan *document frequency* yang relatif besar. Beberapa sampel *query* yang dipakai adalah:

- Pantai
- Gunung
- Candi
- Hotel
- Kuliner
- Jawa
- Bali
- Wisata
- Pulau
- Bukit

Query diatas akan digunakan untuk menganalisa parameter *term frequency* dalam dokumen dan *inverse document frequency*. Hasil akhir analisa adalah nilai tertinggi, terendah, rata-rata, kuartil atas, kuartil bawah. Hasil analisa parameter TF dalam dokumen ditunjukkan pada Tabel 3.3 dan hasil analisa parameter IDF ditunjukkan pada Tabel 3.4.

Tabel 3.3. Hasil Analisa Term Frequency didalam Dokumen

Query	Tertinggi	Terendah	Rata - Rata	Kuartil Atas	Kuartil Bawah
Pantai	0.1913	0.0011	0.0390	0.1152	0.0201
Gunung	0.0857	0.0014	0.0140	0.0499	0.0077
Candi	0.1938	0.0013	0.0220	0.1079	0.0116
Hotel	0.1022	0.0012	0.0090	0.0556	0.0051
Kuliner	0.0743	0.0011	0.0150	0.0447	0.0080
Jawa	0.0658	0.0011	0.0090	0.0374	0.0050
Bali	0.0826	0.0013	0.0170	0.0498	0.0091
Wisata	0.1564	0.0018	0.0320	0.0942	0.0169

Query	Tertinggi	Terendah	Rata - Rata	Kuartil Atas	Kuartil Bawah
Pulau	0.1159	0.0011	0.0200	0.0680	0.0106
Bukit	0.0797	0.0011	0.0070	0.0434	0.0041
Rata-rata	0.1148	0.0012	0.0184	0.0666	0.0098

Tabel 3.4. Hasil Analisa Inverse Document Frequency

Keyword	IDF
Pantai	0.67
Gunung	0.96
Candi	2.11
Hotel	1.12
Kuliner	1.08
Jawa	0.86
Bali	1.57
Wisata	0.04
Pulau	0.86
Bukit	1.52
Rata-rata	1.08
Tertinggi	2.11
Terendah	0.04
Kuartil Bawah	0.56
Kuartil Atas	1.59

Selain *term frequency* didalam dokumen dan *inverse document frequency*, parameter *fuzzy* lain tidak membutuhkan *query* sampel untuk dilakukan analisa. Parameter tersebut adalah *term frequency* didalam *query* yang ditunjukkan pada Tabel 3.5, *inverse document length* yang ditunjukkan pada Tabel 3.6, dan *inverse query length* yang ditunjukkan pada Tabel 3.7.

Tabel 3.5. Hasil Analisa Term Frequency didalam Query

Frekuensi Kemunculan	Panjang Query				
	1	2	3	4	5
1	1	0.50	0.33	0.25	0.20
2	-	1	0.60	0.50	0.40
3	-	-	1	0.75	0.60
4	-	-	-	1	0.80
5	-	-	-	-	1

Tabel 3.6. Hasil Analisa Inverse Document Length

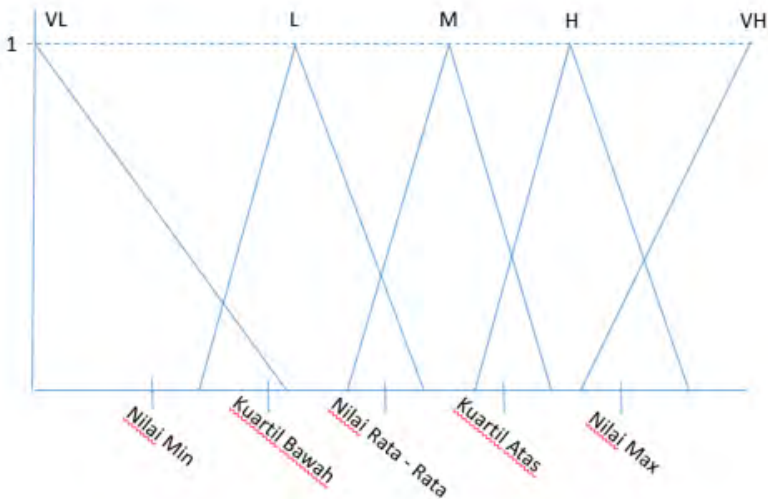
Nilai Tertinggi	0.114
Nilai Terendah	0.024
Rata - rata	0.054
Kuartil Atas	0.039
Kuartil Bawah	0.084

Tabel 3.7. Hasil Analisa Inverse Query Length

Panjang Query	Nilai Norm
1	1.00
2	0.71
3	0.58
4	0.50
5	0.45
6	0.41
7	0.38
8	0.35
9	0.33
10	0.32

3.3.2.2.2. Aturan Penentuan Membership Function

Hasil analisa yang didapatkan pada bagian sebelumnya dapat digunakan sebagai acuan untuk menentukan *membership function*. Penentuan *membership function* sendiri harus dilakukan dengan aturan-aturan yang didasarkan pada hasil analisa dan jumlah kelas *fuzzy*. Dari hasil analisa *term frequency* didalam dokumen, *inverse document frequency*, dan *inverse document length* didapatkan nilai tertinggi, terendah, rata – rata, kuartil atas, kuartil bawah dan parameter – parameter tersebut memiliki 5 kelas *fuzzy* (*very low, low, medium, high, very high*) maka dapat dibuat aturan penentuan *membership function* seperti yang ditunjukkan pada Gambar 3.13.



Gambar 3.13. Aturan Penentuan Membership Function

Pada gambar diatas dijelaskan penentuan *membership function* untuk tiap-tiap kelas. Kelas *very low* memiliki batas bawahnya merupakan nilai terendah yang mungkin didapatkan parameter tersebut dan batas atas berada di depan kuartil bawah. Kelas *low* memiliki batas bawah di belakang kuartil bawah dan batas atas di depan nilai rata-rata. Kelas *medium* memiliki batas

bawah di belakang nilai rata-rata dan batas atas di depan nilai kuartil atas. Kelas *high* memiliki batas bawah di belakang kuartil atas dan batas atas di depan nilai tertinggi. Kelas *very high* memiliki batas bawah di belakang nilai tertinggi dan batas atasnya merupakan nilai tertinggi yang mungkin didapatkan parameter tersebut. Untuk parameter *term frequency* di dalam *query* dan *inverse query length*, penentuan *membership function* dapat dilakukan dengan melihat kecenderungan nilai yang dihasilkan karena hasil analisisnya tidak menghasilkan nilai terendah, tertinggi, rata-rata, kuartil bawah, dan kuartil atas.

3.3.2.2.3. Membership Function Term Frequency di dalam Dokumen

Hasil analisa yang ditunjukkan pada Tabel 3.3 dan aturan yang ditunjukkan pada *Gambar 3.13* dapat digunakan sebagai dasar untuk menentukan *membership function* dari parameter *term frequency* didalam dokumen.

Untuk kelas *very low* akan diambil batas bawah 0 dan batas atas 0.01. Penentuan batas bawah dilakukan dengan mengambil nilai terendah yang mungkin didapatkan oleh parameter tersebut yaitu 0, sedangkan penentuan batas atas dilakukan dengan melakukan pengambilan nilai yang ada di antara nilai kuartil bawah (0.009) dan nilai rata-rata (0.018).

Untuk kelas *low* akan diambil batas bawah 0.005 dan batas atas 0.02. Penentuan batas bawah dilakukan dengan melakukan pengambilan nilai yang ada di antara nilai terendah (0.001) dan nilai kuartil bawah (0.009), sedangkan penentuan batas atas dilakukan dengan melakukan pengambilan nilai yang ada di antara nilai rata-rata (0.018) dan nilai kuartil atas (0.066).

Untuk kelas *medium* akan diambil batas bawah 0.015 dan batas atas 0.075. Penentuan batas bawah dilakukan dengan melakukan pengambilan nilai yang ada di antara nilai kuartil bawah (0.009) dan nilai rata-rata (0.018), sedangkan penentuan batas atas dilakukan dengan melakukan pengambilan nilai yang ada di antara nilai kuartil atas (0.066) dan nilai tertinggi (0.114).

Untuk kelas *high* akan diambil batas bawah 0.06 dan batas atas 0.12. Penentuan batas bawah dilakukan dengan melakukan pengambilan nilai yang ada di antara nilai rata-rata (0.018) dan nilai kuartil atas (0.066), sedangkan penentuan batas atas dilakukan dengan melakukan pengambilan nilai yang ada di antara nilai tertinggi (0.114) dan nilai nilai tertinggi yang mungkin didapatkan parameter (1).

Untuk kelas *very high* akan diambil batas bawah 0.1 dan batas atas 1. Penentuan batas bawah dilakukan dengan melakukan pengambilan nilai yang ada di antara nilai kuartil atas (0.066) dan nilai tertinggi (0.114), sedangkan penentuan batas atas dilakukan dengan mengambil nilai tertinggi yang mungkin didapatkan oleh parameter (1).

Dari hasil penentuan di atas, maka *membership function* untuk parameter *term frequency* di dalam dokumen dapat ditunjukkan seperti pada Tabel 3.8.

Tabel 3.8. Membership Function Term Frequency di dalam Dokumen

Parameter	Kelas Fuzzy	Fungsi Keanggotaan
TF	Very Low	$TF - VL(x) = \begin{cases} 0, & x \geq 0.01 \\ \frac{0.01 - x}{0.01}, & 0 < x < 0.01 \\ 1, & x \leq 0 \end{cases}$
	Low	$TF - L(x) = \begin{cases} 0, & x \leq 0.005 \text{ atau } x \geq 0.02 \\ \frac{x - 0.005}{0.0075}, & 0.005 < x \leq 0.0125 \\ \frac{0.2 - x}{0.0075}, & 0.0125 \leq x < 0.02 \end{cases}$
	Medium	$TF - M(x) = \begin{cases} 0, & x \leq 0.015 \text{ atau } x \geq 0.075 \\ \frac{x - 0.015}{0.03}, & 0.015 < x \leq 0.045 \\ \frac{0.75 - x}{0.03}, & 0.045 \leq x < 0.075 \end{cases}$

Parameter	Kelas Fuzzy	Fungsi Keanggotaan
	High	$TF - H(x) = \begin{cases} 0, & x \leq 0.06 \text{ atau } x \geq 0.12 \\ \frac{x - 0.06}{0.03}, & 0.06 < x \leq 0.09 \\ \frac{0.12 - x}{0.03}, & 0.09 \leq x < 0.12 \end{cases}$
	Very High	$TF - VH(x) = \begin{cases} 0, & x \leq 0.1 \\ \frac{x - 0.1}{0.9}, & 0.1 < x \leq 1 \\ 1, & x \geq 1 \end{cases}$

3.3.2.2.4. Membership Function Term Frequency di dalam Query

Hasil analisa yang ditunjukkan pada Tabel 3.5 dapat digunakan sebagai dasar untuk menentukan *membership function* dari parameter *term frequency* di dalam *query*. Karena hasil analisa tidak menghasilkan nilai tertinggi, terendah, rata-rata, kuartil atas dan kuartil bawah, maka perlu dilakukan pendekatan lain untuk menentukan *membership function*.

Tabel analisa menunjukkan nilai *term frequency* dari setiap dokumen dengan panjang dan nilai frekuensi berbeda. Frekuensi sebuah *term* di dalam *query* memiliki kecenderungan hanya bernilai 1 – 2. Sangat jarang ditemukan frekuensi yang lebih dari itu. Asumsi diatas dapat digunakan untuk menentukan *membership function*.

Asumsi kemunculan sebuah *term* hanya memiliki frekuensi hanya antara 1 – 2 pada setiap *query* memberikan nilai *term frequency* dengan *range* nilai 0 – 0.6. *Range* tersebut merupakan *range* nilai yang banyak kemungkinan untuk muncul dan dari *range* tersebut dapat ditentukan batas atas untuk tiap kelasnya. Batas atas ditentukan dengan membagi *range* menjadi 4 bagian yang sama (batas atas kelas *very high* merupakan nilai tertinggi yang mungkin didapatkan oleh parameter) sehingga didapatkan batas atas untuk kelas *very low* adalah 0.15, kelas *low* adalah 0.3, kelas *medium* adalah 0.45, kelas *high* adalah 0.6, dan kelas *very high* adalah 1.

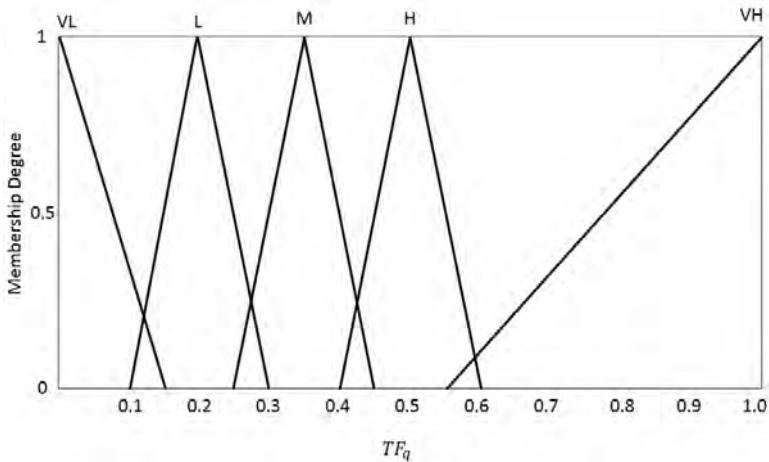
Kelas *low*, *medium*, dan *high* diberikan jarak 0.2 antara batas atas dan batas bawahnya, sedangkan batas bawah untuk kelas *very low* merupakan nilai terendah yang mungkin didapatkan parameter. Batas bawah untuk kelas *very low* adalah 0, kelas *low* adalah 0.1, kelas *medium* adalah 0.25, kelas *high* adalah 0.4, dan kelas *very high* adalah 0.55.

Dari hasil penentuan di atas, maka *membership function* untuk parameter *term frequency* di dalam *query* dapat ditunjukkan seperti pada Tabel 3.9 dan representasi dalam bentuk gambar ditunjukkan pada *Gambar 3.14*.

Tabel 3.9. Membership Function Term Frequency di dalam Query

Parameter	Kelas Fuzzy	Fungsi Keanggotaan
TF	Very Low	$TF - VL(x) = \begin{cases} 0, & x \geq 0.15 \\ \frac{0.15 - x}{0.15}, & 0 < x < 0.15 \\ 1, & x \leq 0 \end{cases}$
	Low	$TF - L(x) = \begin{cases} 0, & x \leq 0.1 \text{ atau } x \geq 0.3 \\ \frac{x - 0.1}{0.1}, & 0.1 < x \leq 0.2 \\ \frac{0.3 - x}{0.1}, & 0.2 \leq x < 0.3 \end{cases}$
	Medium	$TF - M(x) = \begin{cases} 0, & x \leq 0.25 \text{ atau } x \geq 0.45 \\ \frac{x - 0.25}{0.1}, & 0.25 < x \leq 0.35 \\ \frac{0.45 - x}{0.1}, & 0.35 \leq x < 0.45 \end{cases}$
	High	$TF - H(x) = \begin{cases} 0, & x \leq 0.4 \text{ atau } x \geq 0.6 \\ \frac{x - 0.4}{0.1}, & 0.4 < x \leq 0.5 \\ \frac{0.6 - x}{0.1}, & 0.5 \leq x < 0.6 \end{cases}$

Parameter	Kelas Fuzzy	Fungsi Keanggotaan
	Very High	$TF - VH(x) = \begin{cases} 0, & x \leq 0.55 \\ \frac{x - 0.55}{0.45}, & 0.55 < x \leq 1 \\ 1, & x \geq 1 \end{cases}$



**Gambar 3.14. Membership Function Term Frequency
Didalam Query**

3.3.2.2.5. Membership Function Inverse Document Frequency

Hasil analisa yang ditunjukkan pada Tabel 3.4 dan aturan yang ditunjukkan pada *Gambar 3.13* dapat digunakan sebagai dasar untuk menentukan *membership function* dari parameter *inverse document frequency*.

Untuk kelas *very low* akan diambil batas bawah 0 dan batas atas 0.7. Penentuan batas bawah dilakukan dengan mengambil nilai terendah yang mungkin didapatkan oleh parameter tersebut yaitu 0, sedangkan penentuan batas atas dilakukan dengan melakukan pengambilan nilai yang ada di antara nilai kuartil bawah (0.56) dan nilai rata-rata (1.07).

Untuk kelas *low* akan diambil batas bawah 0.4 dan batas atas 1.1. Penentuan batas bawah dilakukan dengan melakukan pengambilan nilai yang ada di antara nilai terendah (0.04) dan nilai kuartil bawah (0.56), sedangkan penentuan batas atas dilakukan dengan melakukan pengambilan nilai yang ada di antara nilai rata-rata (1.07) dan nilai kuartil atas (1.59).

Untuk kelas *medium* akan diambil batas bawah 0.9 dan batas atas 1.7. Penentuan batas bawah dilakukan dengan melakukan pengambilan nilai yang ada di antara nilai kuartil bawah (0.56) dan nilai rata-rata (1.07), sedangkan penentuan batas atas dilakukan dengan melakukan pengambilan nilai yang ada di antara nilai kuartil atas (1.59) dan nilai tertinggi (2.10).

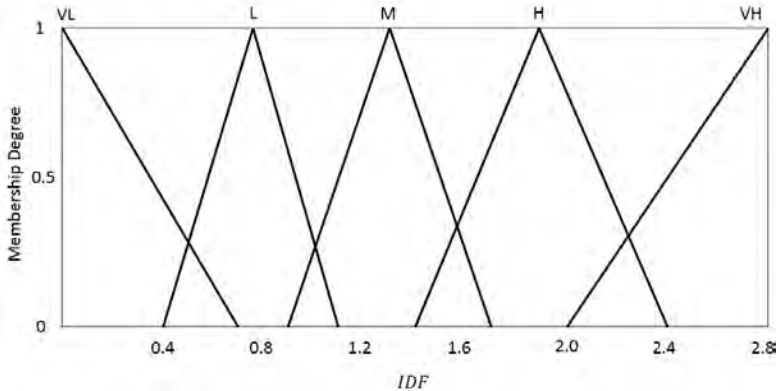
Untuk kelas *high* akan diambil batas bawah 1.4 dan batas atas 2.4. Penentuan batas bawah dilakukan dengan melakukan pengambilan nilai yang ada di antara nilai rata-rata (1.07) dan nilai kuartil atas (1.59), sedangkan penentuan batas atas dilakukan dengan melakukan pengambilan nilai yang ada di antara nilai tertinggi (2.10) dan nilai nilai tertinggi yang mungkin didapatkan parameter (2.8).

Untuk kelas *very high* akan diambil batas bawah 2 dan batas atas 2.8. Penentuan batas bawah dilakukan dengan melakukan pengambilan nilai yang ada di antara nilai kuartil atas (1.59) dan nilai tertinggi (2.10), sedangkan penentuan batas atas dilakukan dengan mengambil nilai tertinggi yang mungkin didapatkan oleh parameter (2.8).

Dari hasil penentuan di atas, maka *membership function* untuk parameter *inverse document frequency* dapat ditunjukkan seperti pada Tabel 3.10 dan representasi gambarnya ditunjukkan pada Gambar 3.15.

Tabel 3.10. Membership Function Inverse Document Frequency

Parameter	Kelas Fuzzy	Fungsi Keanggotaan
IDF	Very Low	$IDF - VL(x) = \begin{cases} 0, & x \geq 0.7 \\ \frac{0.7 - x}{0.7}, & 0 < x < 0.7 \\ 1, & x \leq 0 \end{cases}$
	Low	$IDF - L(x) = \begin{cases} 0, & x \leq 0.4 \text{ atau } x \geq 1.1 \\ \frac{x - 0.4}{0.35}, & 0.4 < x \leq 0.75 \\ \frac{1.1 - x}{0.35}, & 0.75 \leq x < 1.1 \end{cases}$
	Medium	$IDF - M(x) = \begin{cases} 0, & x \leq 0.9 \text{ atau } x \geq 1.7 \\ \frac{x - 0.9}{0.4}, & 0.9 < x \leq 1.3 \\ \frac{1.7 - x}{0.4}, & 1.3 \leq x < 1.7 \end{cases}$
	High	$IDF - H(x) = \begin{cases} 0, & x \leq 1.4 \text{ atau } x \geq 2.4 \\ \frac{x - 1.4}{0.5}, & 1.4 < x \leq 1.9 \\ \frac{2.4 - x}{0.5}, & 1.9 \leq x < 2.4 \end{cases}$
	Very High	$IDF - VH(x) = \begin{cases} 0, & x \leq 2 \\ \frac{x - 2}{0.8}, & 2 < x \leq 2.8 \\ 1, & x \geq 2.8 \end{cases}$



Gambar 3.15. Membership Function Inverse Document Frequency

3.3.2.2.6. Membership Function Inverse Document Length

Hasil analisa yang ditunjukkan pada Tabel 3.6 dan aturan yang ditunjukkan pada *Gambar 3.13* dapat digunakan sebagai dasar untuk menentukan *membership function* dari parameter *inverse document length*.

Untuk kelas *very low* akan diambil batas bawah 0 dan batas atas 0.04. Penentuan batas bawah dilakukan dengan mengambil nilai terendah yang mungkin didapatkan oleh parameter tersebut yaitu 0, sedangkan penentuan batas atas dilakukan dengan melakukan pengambilan nilai yang ada di antara nilai kuartil bawah (0.038) dan nilai rata-rata (0.054).

Untuk kelas *low* akan diambil batas bawah 0.03 dan batas atas 0.06. Penentuan batas bawah dilakukan dengan melakukan pengambilan nilai yang ada di antara nilai terendah (0.023) dan nilai kuartil bawah (0.038), sedangkan penentuan batas atas dilakukan dengan melakukan pengambilan nilai yang ada di antara nilai rata-rata (0.054) dan nilai kuartil atas (0.084).

Untuk kelas *medium* akan diambil batas bawah 0.05 dan batas atas 0.09. Penentuan batas bawah dilakukan dengan melakukan pengambilan nilai yang ada di antara nilai kuartil bawah (0.038) dan nilai rata-rata (0.054), sedangkan penentuan

batas atas dilakukan dengan melakukan pengambilan nilai yang ada di antara nilai kuartil atas (0.084) dan nilai tertinggi (0.114).

Untuk kelas *high* akan diambil batas bawah 0.08 dan batas atas 0.12. Penentuan batas bawah dilakukan dengan melakukan pengambilan nilai yang ada di antara nilai rata - rata (0.054) dan nilai kuartil atas (0.084), sedangkan penentuan batas atas dilakukan dengan melakukan pengambilan nilai yang ada di antara nilai tertinggi (0.114) dan nilai nilai tertinggi yang mungkin didapatkan parameter (1).

Untuk kelas *very high* akan diambil batas bawah 0.11 dan batas atas 1. Penentuan batas bawah dilakukan dengan melakukan pengambilan nilai yang ada di antara nilai kuartil atas (0.084) dan nilai tertinggi (0.114), sedangkan penentuan batas atas dilakukan dengan mengambil nilai tertinggi yang mungkin didapatkan oleh parameter (1).

Dari hasil penentuan di atas, maka *membership function* untuk parameter *inverse document length* dapat ditunjukkan seperti pada Tabel 3.11

Tabel 3.11. Membership Function Inverse Document Length

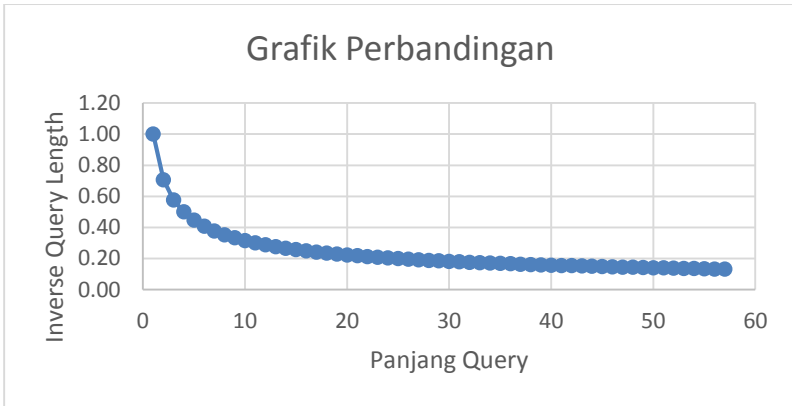
Parameter	Kelas Fuzzy	Fungsi Keanggotaan
N	Very Low	$N - VL(x) = \begin{cases} 0, & x \geq 0.04 \\ \frac{0.04 - x}{0.04}, & 0 < x \leq 0.04 \\ 1, & x \leq 0 \end{cases}$
	Low	$N - L(x) = \begin{cases} 0, & x \leq 0.03 \text{ atau } x \geq 0.06 \\ \frac{x - 0.03}{0.015}, & 0.03 < x \leq 0.045 \\ \frac{0.06 - x}{0.015}, & 0.045 \leq x < 0.06 \end{cases}$

Parameter	Kelas Fuzzy	Fungsi Keanggotaan
	Medium	$N - M(x) = \begin{cases} 0, & x \leq 0.05 \text{ atau } x \geq 0.09 \\ \frac{x - 0.05}{0.02}, & 0.05 < x \leq 0.07 \\ \frac{0.09 - x}{0.02}, & 0.07 \leq x < 0.09 \end{cases}$
	High	$N - H(x) = \begin{cases} 0, & x \leq 0.08 \text{ atau } x \geq 0.12 \\ \frac{x - 0.08}{0.02}, & 0.08 < x \leq 0.1 \\ \frac{0.12 - x}{0.02}, & 0.1 \leq x < 0.12 \end{cases}$
	Very High	$N - VH(x) = \begin{cases} 0, & x \leq 0.11 \\ \frac{x - 0.11}{0.89}, & 0.11 < x \leq 1 \\ 1, & x \geq 1 \end{cases}$

3.3.2.2.7. Membership Function Inverse Query Length

Hasil analisa yang ditunjukkan pada Tabel 3.7 dapat digunakan sebagai dasar untuk menentukan *membership function* dari parameter *inverse query length*. Karena hasil analisa tidak menghasilkan nilai tertinggi, terendah, rata-rata, kuartil atas dan kuartil bawah, maka perlu dilakukan pendekatan lain untuk menentukan *membership function*.

Hasil analisa menunjukkan penurunan nilai *inverse query length* yang semakin berkurang dengan bertambahnya panjang *query*. Selisih nilai *inverse query length* dari *query* dengan panjang 1 dan 2 adalah 0.29 sedangkan selisih nilai *inverse query length* dari *query* dengan panjang 2 dan 3 adalah 0.13. Grafik penurunan nilai dari *inverse query length* ditunjukkan pada Gambar 3.16.



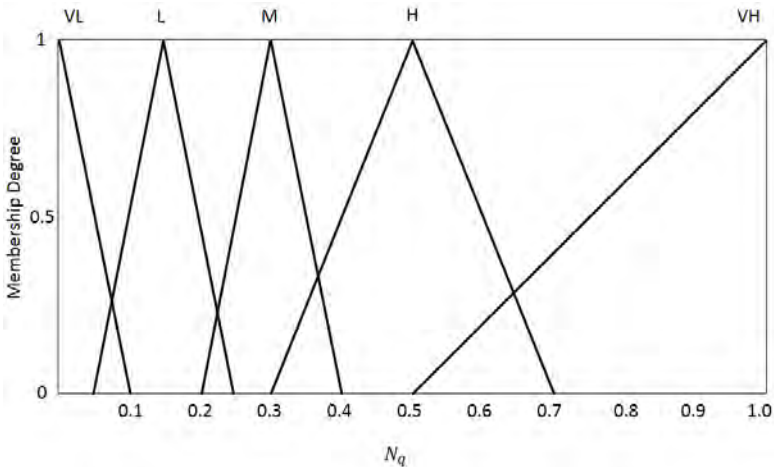
Gambar 3.16. Grafik Perbandingan Nilai Inverse Query Length dengan Panjang Query

Grafik diatas menunjukkan bahwa nilai *inverse query length* banyak tersebar pada range 0.1 – 0.4 dan lebih sedikit pada range 1 – 0.4. Maka cakupan kelas pada *membership function* harus difokuskan pada daerah mayoritas. *Membership function* untuk parameter *inverse query length* dapat ditunjukkan seperti pada Tabel 3.12 dan representasi gambarnya ditunjukkan dalam Gambar 3.17.

Tabel 3.12. Membership Function Inverse Query Length

Parameter	Kelas Fuzzy	Fungsi Keanggotaan
N	Very Low	$N - VL(x) = \begin{cases} 0, & x \geq 0.1 \\ \frac{0.1 - x}{0.1}, & 0 < x \leq 0.1 \\ 1, & x \leq 0 \end{cases}$

Parameter	Kelas Fuzzy	Fungsi Keanggotaan
	Low	$N - L(x) = \begin{cases} 0, & x \leq 0.05 \text{ atau } x \geq 0.25 \\ \frac{x - 0.05}{0.1}, & 0.05 < x \leq 0.15 \\ \frac{0.25 - x}{0.1}, & 0.15 \leq x < 0.25 \end{cases}$
	Medium	$N - M(x) = \begin{cases} 0, & x \leq 0.2 \text{ atau } x \geq 0.4 \\ \frac{x - 0.2}{0.1}, & 0.2 < x \leq 0.3 \\ \frac{0.4 - x}{0.1}, & 0.3 \leq x < 0.4 \end{cases}$
	High	$N - H(x) = \begin{cases} 0, & x \leq 0.3 \text{ atau } x \geq 0.7 \\ \frac{x - 0.3}{0.2}, & 0.3 < x \leq 0.5 \\ \frac{0.7 - x}{0.2}, & 0.5 \leq x < 0.7 \end{cases}$
	Very High	$N - VH(x) = \begin{cases} 0, & x \leq 0.5 \\ \frac{x - 0.5}{0.5}, & 0.5 < x \leq 1 \\ 1, & x \geq 1 \end{cases}$



Gambar 3.17. Membership Function Inverse Query Length

3.3.2.2. Penentuan Fuzzy Rule

Di bagian ini akan dilakukan penentuan *rule-rule* yang akan digunakan dalam algoritma *fuzzy*. Jumlah *fuzzy rule* yang mungkin pada sebuah *fuzzy logic controller* bergantung pada jumlah parameter yang diterima beserta jumlah kelas *fuzzy* dari parameter-parameter tersebut. Jumlah *fuzzy rule* yang mungkin pada setiap *fuzzy logic controller* ditunjukkan pada Tabel 3.13.

Tabel 3.13. Jumlah Kemungkinan Fuzzy Rule

Level	FLC	Parameter (Jumlah Kelas)	Kemungkinan Rule
1	FLC Document	TF_d (5)	$5 * 5 * 5 = 125$
		IDF (5)	
		N_d (5)	
	FLC Query	TF_q (5)	

Level	FLC	Parameter (Jumlah Kelas)	Kemungkinan Rule
		IDF (5)	$5 * 5 * 5 = 125$
		N_q (5)	
2	FLC Output	W_{id} (3)	$3 * 3 = 9$
		W_{tq} (3)	

Tabel diatas menunjukkan terdapt 250 kemungkinan *fuzzy rule* pada tingkat pertama dan 9 kemungkinan pada tingkat kedua. Total seluruh kemungkinan terdapat 259 *rule*. Beberapa *rule* dapat digabungkan karena menghasilkan *conclusion* yang sama, contoh proses penggabungan *rule* ini ditunjukkan pada *Gambar 3.18*. *Rule-rule* yang akan digunakan ditunjukkan pada Tabel 3.14 untuk level pertama dan .Tabel 3.15 untuk level kedua.

Pembangunan *rule-rule* ini didasarkan pada diagram yang ditunjukkan pada *Gambar 2.5*. Daerah dari tiap parameter akan dibagi sama rata sesuai jumlah kelasnya. Setelah pembagian daerah sesuai dengan kelasnya telah dilakukan, proses selanjutnya adalah memetakan masing-masing kelas parameter ke dalam diagram untuk mendapatkan *conclusion* dari *rule fuzzy*.

Rule 1 = IF tf Low AND idf Low AND norm high THEN weight Medium

Rule 2 = IF tf Very Low AND idf Low AND norm very high THEN weight Medium

Rule Gabungan = IF tf Very Low OR Low AND idf Low AND norm very high OR high THEN weight Medium

Gambar 3.18. Contoh Proses Penggabungan Rule

Tabel 3.14. Rule Tingkat Pertama

No	TF	IDF	N	THEN	W
1	Low OR VeryLow	Low OR VeryLow	Low OR VeryLow	THEN	Low

2	Low OR VeryLow	Low OR VeryLow	Medium	THEN	Low
3	Low OR VeryLow	Low OR VeryLow	High Or VeryHigh	THEN	Low
4	Low OR VeryLow	Medium	Low OR VeryLow	THEN	Medium
5	Low OR VeryLow	Medium	Medium	THEN	Medium
6	Low OR VeryLow	Medium	High Or VeryHigh	THEN	Medium
7	Low OR VeryLow	High Or VeryHigh	Low OR VeryLow	THEN	Medium
8	Low OR VeryLow	High Or VeryHigh	Medium	THEN	Medium
9	Low OR VeryLow	High Or VeryHigh	High Or VeryHigh	THEN	Medium
10	Medium	Low OR VeryLow	Low OR VeryLow	THEN	Medium
11	Medium	Low OR VeryLow	Medium	THEN	Medium
12	Medium	Low OR VeryLow	High Or VeryHigh	THEN	Medium
13	Medium	Medium	Low OR VeryLow	THEN	Medium
14	Medium	Medium	Medium	THEN	Medium
15	Medium	Medium	High Or VeryHigh	THEN	Medium
16	Medium	High Or VeryHigh	Low OR VeryLow	THEN	Medium
17	Medium	High Or VeryHigh	Medium	THEN	Medium
18	Medium	High Or VeryHigh	High Or VeryHigh	THEN	Medium
19	High Or VeryHigh	Low OR VeryLow	Low OR VeryLow	THEN	Medium

20	High Or VeryHigh	Low OR VeryLow	Medium	THEN	Medium
21	High Or VeryHigh	Low OR VeryLow	High Or VeryHigh	THEN	Medium
22	High Or VeryHigh	Medium	Low OR VeryLow	THEN	Medium
23	High Or VeryHigh	Medium	Medium	THEN	Medium
24	High Or VeryHigh	Medium	High Or VeryHigh	THEN	Medium
25	High Or VeryHigh	High Or VeryHigh	Low OR VeryLow	THEN	Medium
26	High Or VeryHigh	High Or VeryHigh	Medium	THEN	Medium
27	High Or VeryHigh	High Or VeryHigh	High Or VeryHigh	THEN	High

Tabel 3.15. Rule Tingkat Kedua

No	W _{td}	W _{tq}	THEN	Score
1	Low	Low	THEN	Low
2	Low	Medium	THEN	Low
3	Low	High	THEN	Medium
4	Medium	Low	THEN	Low
5	Medium	Medium	THEN	Medium
6	Medium	High	THEN	Medium
7	High	Low	THEN	Medium
8	High	Medium	THEN	Medium
9	High	High	THEN	High

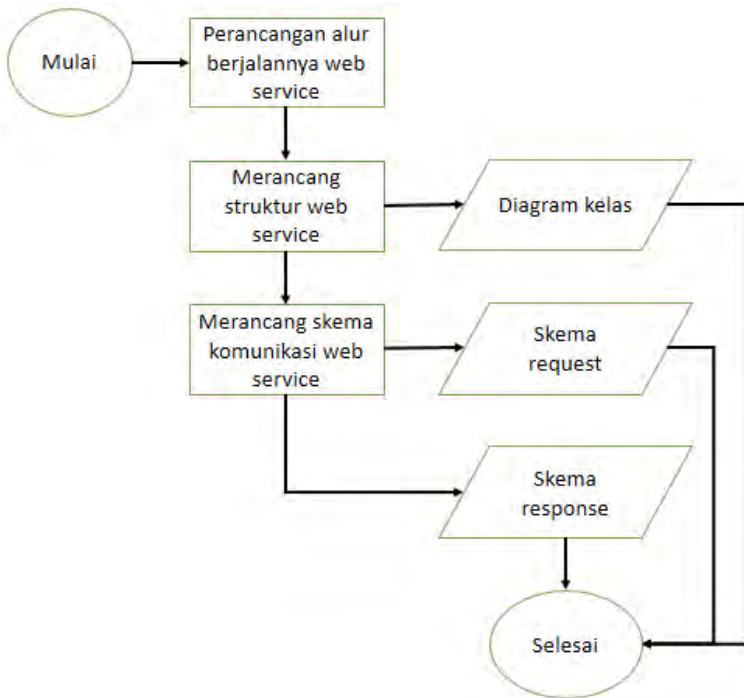
3.3.3. Tahap Perancangan Web Service

Di dalam tahap perancangan *web service* ini terdapat beberapa proses yang dilakukan antara lain analisis implementasi *web service* secara umum, perancangan struktur *web service* dan

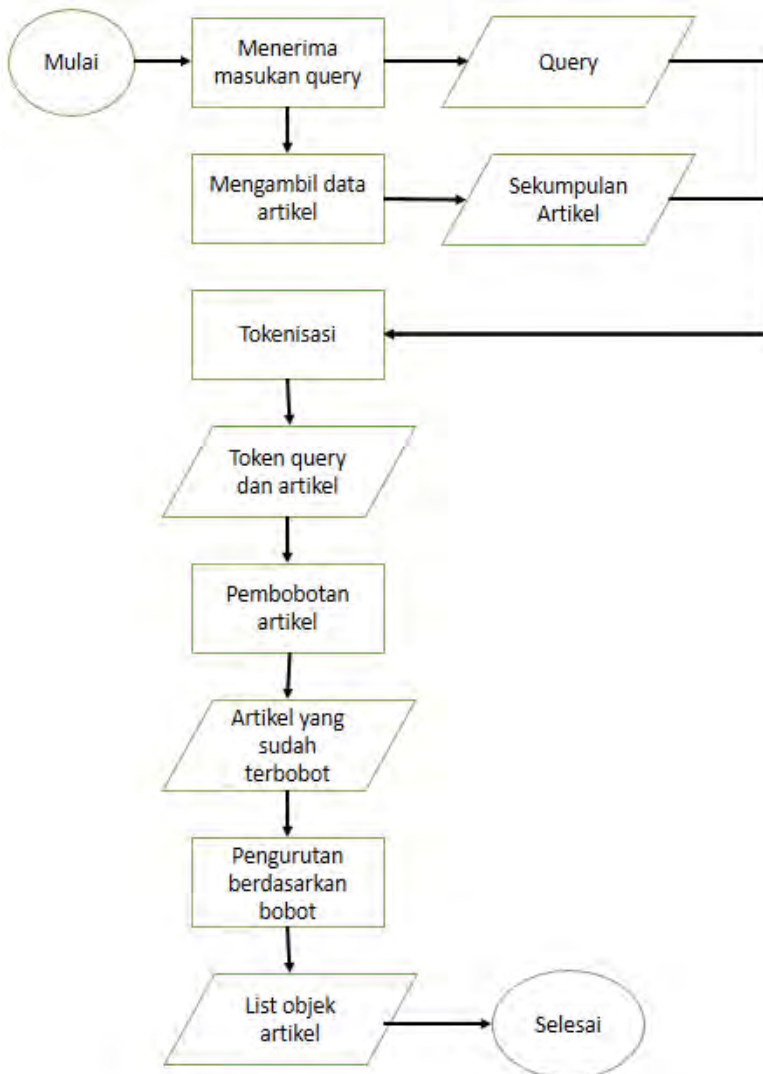
perancangan skema *request* serta *response web service*. Diagram alir mengenai tahap perancangan *web service* ini dapat dilihat pada Gambar 3.19.

3.3.3.1. Analisis Implementasi Web Service Secara Umum

Pada proses ini akan dilakukan analisa proses berjalannya *web service*. *Web service* akan menerima sebuah masukan *query* yang berbentuk *string*. Setelah menerima masukan *query*, proses selanjutnya adalah mengambil artikel–artikel yang tersimpan di *database*. Data *query* dan artikel yang telah diterima kemudian akan ditokenisasi agar dapat diproses ke tahap berikutnya. *Query* dan artikel yang sudah berbentuk token selanjutnya akan dilakukan proses pembobotan sesuai dengan algoritma yang telah ditentukan. Setelah proses pembobotan sudah dilakukan, setiap artikel akan memiliki nilai yang menyatakan relevansinya dengan masukan *query*. Proses selanjutnya adalah melakukan pengurutan artikel berdasarkan hasil pembobotan dan menyimpan artikel–artikel itu kedalam sebuah objek yang siap dikirimkan sebagai keluaran dari *web service*. Keluaran *web service* berupa sekumpulan objek artikel yang terurut berdasarkan bobotnya. Diagram alir proses berjalannya *web service* ditunjukkan pada Gambar 3.20.



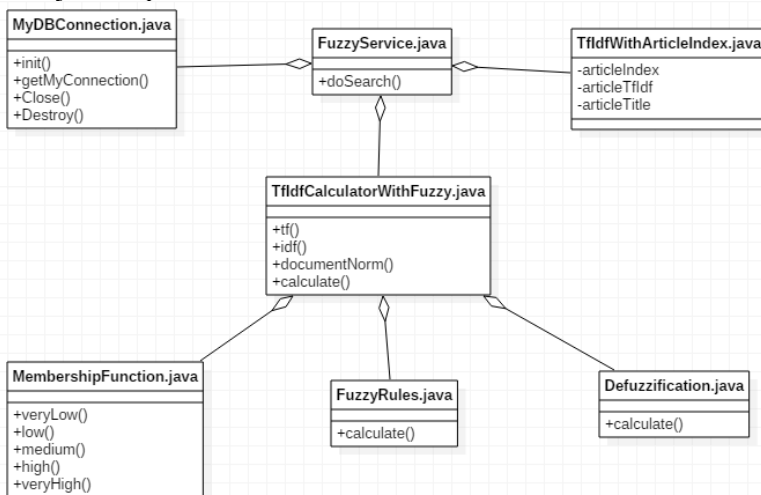
Gambar 3.19. Diagram Alir Tahap Perancangan Web Service



Gambar 3.20. Diagram Alir Berjalannya Web Service

3.3.3.2. Perancangan Struktur Web Service

Pada proses ini akan dilakukan proses perancangan diagram kelas dari *web service* yang selanjutnya akan digunakan sebagai acuan untuk proses implementasi *web service*. Pembuatan diagram kelas ini didasarkan pada alur program yang ditunjukkan pada Gambar 3.20 dan algoritma yang telah dirancang pada Tahap perancangan algoritma *fuzzy*. Diagram kelas *web service* ditunjukkan pada Gambar 3.21.



Gambar 3.21. Diagram Kelas Web Service

Seluruh kelas beserta atribut dan operasi akan dijelaskan sebagai berikut:

- **FuzzyService**: Kelas utama
 - Operasi
 - `doSearch()` : Operasi yang mengimplementasikan diagram alir pada Gambar 3.20.
- **MyDBConnection**: Kelas yang menampung seluruh operasi *database*
 - Atribut

- Connection : objek koneksi *database*
- Operasi
 - Init() : Operasi untuk inialisasi koneksi ke *database*
 - getMyConnection() : Operasi untuk mendapatkan objek koneksi
 - close() : Operasi untuk menutup operasi *database* yang sedang dilakukan
 - destroy() : Operasi untuk menutup koneksi ke *database*
- TfIdfWithArticleIndex : Kelas untuk menyimpan objek tiap artikel yang dibuat berdasarkan data keluaran seperti yang ditunjukkan pada Tabel 3.2.
 - Atribut
 - articleIndex : index artikel
 - articleTfIdf : *score* artikel
 - articleTitle : judul artikel
 - Operasi
 - getArticleIndex() : Operasi untuk mendapatkan index artikel
 - getArticleTfIdf() : Operasi untuk mendapatkan *score* artikel
 - getArticleTitle() : Operasi untuk mendapatkan judul artikel
 - compareWith() : Operasi untuk membandingkan objek sekarang dengan objek lain
- TfIdfCalculatorFuzzy: Kelas yang digunakan untuk melakukan operasi *fuzzy* seperti yang ditunjukkan pada contoh perhitungan di Tabel 2.2
 - Operasi
 - tf() : Operasi untuk melakukan perhitungan TF (Term Frequency)

- `idf()` : Operasi untuk melakukan perhitungan IDF (*Inverse Document Frequency*)
 - `normDocument()` : Operasi untuk melakukan perhitungan *Norm Document*
 - `calculate()` : Operasi utama yang melakukan perhitungan *fuzzy*
- **MembershipFunction** : Kelas yang digunakan untuk mengubah nilai *crisp* menjadi nilai *fuzzy*. Fungsi ini mengimplementasikan beberapa *membership function* seperti yang ditunjukkan pada Tabel 3.8 hingga Tabel 3.12.
 - Operasi
 - `veryLow()` : Operasi untuk mendapatkan nilai *fuzzy* pada kelas very low
 - `low()` : Operasi untuk mendapatkan nilai *fuzzy* pada kelas low
 - `medium()` : Operasi untuk mendapatkan nilai *fuzzy* pada kelas medium
 - `high()` : Operasi untuk mendapatkan nilai *fuzzy* pada kelas high
 - `veryHigh()` : Operasi untuk mendapatkan nilai *fuzzy* pada kelas very high
- **FuzzyRules** : Kelas yang digunakan untuk melakukan operasi *rule – rule fuzzy*. *Rule-rule* yang dipakai ditunjukkan pada Tabel 3.14 dan Tabel 3.15.
 - Operasi
 - `calculate()` : Operasi untuk melakukan perhitungan *fuzzy rule*
 - `zValue()` : Operasi yang digunakan untuk mengubah nilai *fuzzy* menjadi nilai *crisp*
- **Defuzzification** : Kelas yang digunakan untuk melakukan defuzifikasi. Defuzifikasi ini dilakukan sesuai dengan persamaan yang ditunjukkan pada Persamaan 2.4

- Operasi
 - `calculate()` : Operasi yang digunakan untuk menghitung hasil defuzifikasi

3.3.3.3. Perancangan Skema Request dan Response Web Service

Pada proses ini akan dilakukan perancangan skema *request* yang akan diterima *web service* dan skema *request* yang akan dikirimkan oleh *web service*. Skema *request* untuk *web service* hanya berupa sebuah string *query*, sedangkan skema *response web service* berupa sekumpulan objek artikel yang memiliki atribut id artikel, bobot artikel, judul artikel dan isi artikel. Skema *request* dan *response web service* ditunjukkan pada Tabel 3.16 serta representasinya dalam bentuk XML ditunjukkan pada Gambar 3.22 untuk skema *request* dan Gambar 3.23 untuk skema *response*.

Tabel 3.16. Skema Request dan Response Web Service

	Atribut	Tipe Data	Keterangan
Request	Query	String	<i>Query</i> yang digunakan untuk pembobotan artikel
Response	articleIndex	Int	Index artikel didalam <i>database</i>
	articleTfIdf	Double	Bobot artikel
	articleTitle	String	Judul artikel
	articleContent	String	Isi artikel

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <S:Body>
    <ns2:doSearch xmlns:ns2="http://service.fuzzy.com/">
      <query>bukit dieng</query>
    </ns2:doSearch>
  </S:Body>
</S:Envelope>
```

Gambar 3.22. Contoh Skema Request Web Service

```

<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-
ENV:Header/>
<S:Body>
  <ns2:doSearchResponse xmlns:ns2="http://service.fuzy.com/">
    <return xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="ns2:TfidWithArticleIndex">
      <articleIndex>254</articleIndex>
      <articleTfid>0.318118269800506</articleTfid>
      <articleTitle>Wisata Dieng Monosobo</articleTitle>
      <articleContent>Nama â€" dieng â€" yang secara harfiah diterjemahkan s</articleContent>
    </return>
    <return xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="ns2:TfidWithArticleIndex">
      <articleIndex>494</articleIndex>
      <articleTfid>0.2880819193473319</articleTfid>
      <articleTitle>5 Tempat Wisata Di Dieng Monosobo yang Wajib Dikunjungi</articleTitle>
      <articleContent>Dieng adalah salah satu dataran tinggi yang memiliki</articleContent>
    </return>
  </doSearchResponse>
</S:Body>
</SOAP-ENV:Envelope>

```

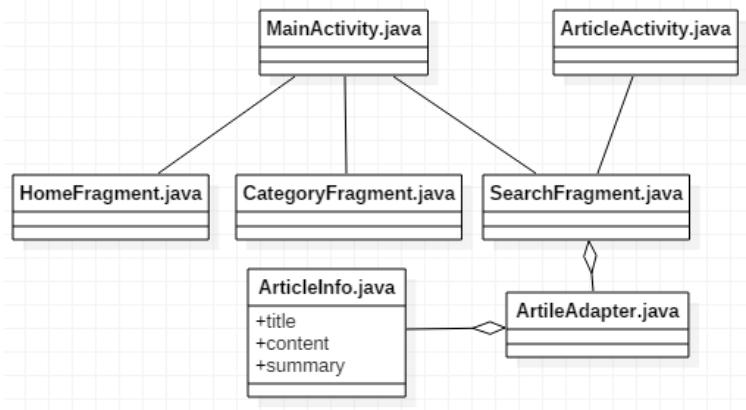
Gambar 3.23. Contoh Skema Response Web Service

3.3.4. Tahap Perancangan Aplikasi Mobile

Setelah *web service* selesai dibuat, tahap selanjutnya yang harus dilakukan adalah membuat aplikasi di perangkat *mobile* yang akan melakukan komunikasi dengan *web service*. Dalam tahap ini hanya dilakukan proses perancangan struktur aplikasi *mobile* yang menghasilkan sebuah diagram kelas.

3.3.4.1. Perancangan Struktur Aplikasi Mobile

Di proses perancangan struktur aplikasi *mobile* ini akan dilakukan perancangan kelas diagram dari aplikasi *mobile* yang akan dibuat. Kelas diagram dari aplikasi ini ditunjukkan pada Gambar 3.24. Penjelasan tiap kelasnya ditunjukkan pada Tabel 3.17.



Gambar 3.24. Diagram Kelas Aplikasi Mobile

Tabel 3.17. Penjelasan Diagram Kelas Aplikasi Mobile

Kelas	Keterangan
MainActivity.java	Kelas utama dari aplikasi <i>mobile</i>
HomeFragment.java	<i>Fragment</i> yang berisi tampilan halaman utama aplikasi

CategoryFragment.java	<i>Fragment</i> yang berisi tampilan halaman kategori
SearchFragment.java	<i>Fragment</i> yang berisi tampilan halaman pencarian
ArticleAdapter.java	Kelas <i>adapter</i> yang digunakan untuk <i>recyclerview</i> pada SearchFragment
ArticleInfo.java	Kelas yang berisi atribut dari objek artikel
ArticleFragment.java	<i>Fragment</i> untuk menampilkan isi artikel
CategoryListFragment.java	Fragment untuk menampilkan daftar artikel berdasar kategori

3.3.4.2. Koneksi Aplikasi Mobile dengan Web Service

Di proses koneksi aplikasi *mobile* dengan *web service* ini akan dilakukan perancangan proses untuk melakukan komunikasi antara aplikasi Android dengan *web service*. Untuk memudahkan proses komunikasi ini maka akan digunakan *library* KSOAP2 yang merupakan *library* Java untuk memudahkan komunikasi dengan *web service* berbasis SOAP. Sebelumnya telah ditunjukkan alur penggunaan *library* KSOAP2 pada Gambar 2.6.

Atribut – atribut koneksi *web service* dapat dilihat pada file WSDL yang dihasilkan oleh *web service*. Atribut – atribut ini berupa *Namespace*, *method name*, *soap action*, dan URL koneksi ke *web service*.

Proses selanjutnya adalah mengirim data yang dibutuhkan oleh *web service* sebagai inputan. Data yang dibutuhkan oleh *web service* ditunjukkan pada Tabel 3.18.

Tabel 3.18. Data Inputan Web Service

Nama Atribut	Type Data
query	String

Setelah atribut ditentukan, maka koneksi ke *web service* dapat dilakukan dengan menggunakan *HttpTransport*. Proses selanjutnya adalah mengambil elemen – elemen yang dibutuhkan dari *response web service*. Elemen – elemen yang akan diambil ditunjukkan pada Tabel 3.19.

Tabel 3.19. Elemen - Elemen yang Diambil Dari Response Web Service

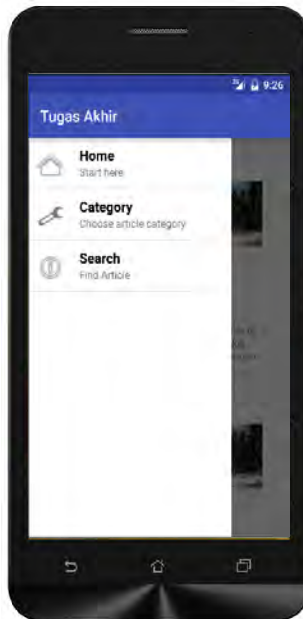
Nama Atribut	Tipe Data	Keterangan
articleTitle	String	Judul Artikel
articleContent	String	Isi Artikel
articleCategory	String	Kategori Artikel

3.4. Perancangan Antarmuka Perangkat Lunak

Pada subbab ini akan dibahas mengenai perancangan antarmuka perangkat lunak yang bertujuan untuk dapat mempermudah interaksi antara perangkat lunak dengan pengguna. Sistem ini memiliki beberapa halaman antara lain yang pertama adalah sebuah *slide drawer* yang digunakan untuk navigasi, halaman utama, halaman pilih kategori, halaman artikel berdasar kategori, halaman pencarian, dan halaman yang menampilkan isi artikel.

3.4.1. Slide Drawer Navigasi

Slide drawer yang digunakan untuk navigasi adalah elemen aplikasi yang ada pada seluruh halaman aplikasi. Elemen ini digunakan untuk membantu pengguna berpindah halaman di dalam aplikasi. Halaman–halaman yang dimaksud adalah halaman utama aplikasi, halaman pemilihan kategori, dan halaman pencarian. Rancangan *slide drawer* navigasi ditunjukkan pada Gambar 3.25 sedangkan penjelasan mengenai atribut yang terdapat pada *slide drawer* ini ditunjukkan pada Tabel 3.20.



Gambar 3.25. Rancangan Slide Drawer Navigasi

Tabel 3.20. Spesifikasi Atribut Antarmuka Slide Drawer Navigasi

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	navList	ListView	Menampilkan daftar menu navigasi ke halaman yang ada di aplikasi	<i>Event Clicked</i>

3.4.2. Halaman Utama

Halaman utama ini adalah halaman pertama yang akan ditampilkan ketika pengguna membuka aplikasi. Pada halaman ini aplikasi akan menampilkan beberapa artikel acak yang ditampilkan dalam sebuah *list*. Rancangan halaman utama ini dapat dilihat pada Gambar 3.26 sedangkan untuk spesifikasi atribut yang ada pada halaman utama ini ditunjukkan oleh Tabel 3.21.



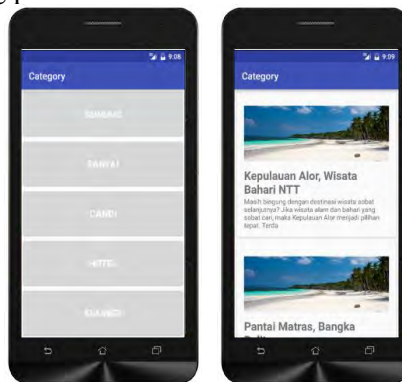
Gambar 3.26. Rancangan Halaman Utama Aplikasi

Tabel 3.21. Spesifikasi Atribut Antarmuka Halaman Utama Aplikasi

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	homeCardList	RecyclerView	Sebuah list yang menampilkan daftar artikel acak	Event Clicked

3.4.3. Halaman Pemilihan Kategori

Halaman ini adalah halaman yang berisi pilihan untuk menampilkan artikel berdasarkan kategori tertentu, Pada halaman ini ditampilkan 5 buah button yang masing–masing mengarah ke halaman yang menampilkan artikel sesuai kategori yang dipilih, Rancangan antarmuka halaman pemilihan kategori dan halaman yang menampilkan artikel berdasar kategori ditunjukkan pada Gambar 3.27 sedangkan untuk spesifikasi atributnya ditunjukkan masing–masing pada Tabel 3.22 dan Tabel 3.23.



Gambar 3.27. Rancangan Halaman Pemilihan Kategori dan Halaman yang Menampilkan Artikel Berdasar Kategori

Tabel 3.22. Spesifikasi Atribut Antarmuka Halaman Pemilihan Kategori

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	buttonGunung	Button	Memilih untuk menampilkan artikel yang masuk kategori gunung	<i>Event Clicked</i>

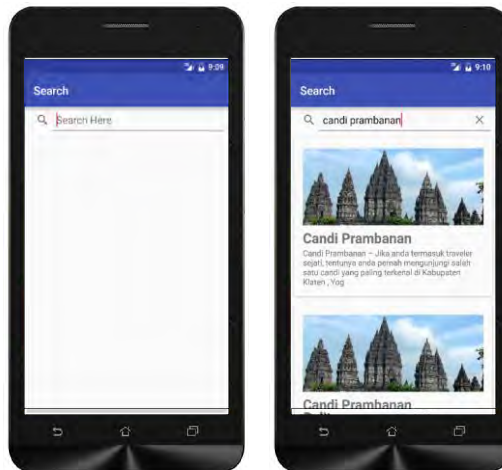
2	buttonPantai	Button	Memilih untuk menampilkan artikel yang masuk kategori pantai	<i>Event Clicked</i>
3	buttonCandi	Button	Memilih untuk menampilkan artikel yang masuk kategori candi	<i>Event Clicked</i>
4	buttonHotel	Button	Memilih untuk menampilkan artikel yang masuk kategori hotel	<i>Event Clicked</i>
5	buttonKuliner	Button	Memilih untuk menampilkan artikel yang masuk kategori kuliner	<i>Event Clicked</i>

Tabel 3.23. Spesifikasi Atribut Antarmuka Halaman Artikel Berdasar Kategori

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	cardListCategory	RecyclerView	Menampilkan artikel berdasarkan kategori yang dipilih	<i>Event Clicked</i>

3.4.4. Halaman Pencarian

Halaman ini adalah halaman yang digunakan untuk melakukan pencarian artikel. Pada halaman ini terdapat sebuah *searchView* yang digunakan untuk memasukkan *query* pencarian, selanjutnya pengguna dapat melakukan pencarian sesuai dengan *query* yang sudah dimasukkan. Hasil pencarian akan ditampilkan pada halaman yang sama dengan menggunakan *recyclerView* yang ada di bagian bawah dari *searchView*. Rancangan halaman pencarian ini ditunjukkan pada Gambar 3.28 sedangkan untuk spesifikasi atributnya ditunjukkan pada Tabel 3.24.



Gambar 3.28. Rancangan Halaman Pencarian

Tabel 3.24. Spesifikasi Atribut Antarmuka Halaman Pencarian

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	doSearch	SearchView	Media bagi pengguna untuk mengisi dan mengirim query yang digunakan untuk melakukan pencarian	<i>Event Clicked</i>
2	cardList	RecyclerView	List yang menampilkan daftar artikel hasil pencarian	<i>Event Clicked</i>

3.4.5. Halaman Detail Artikel

Halaman ini adalah halaman yang menampilkan artikel yang dipilih dari dalam *list* yang ada pada halaman-halaman sebelumnya. Pada halaman ini akan ditampilkan sebuah gambar yang mewakili kategori dari artikel tersebut beserta judul dan isi

dari artikel. Rancangan antarmuka halaman ini dapat dilihat pada Gambar 3.29 sedangkan untuk spesifikasi atributnya dapat dilihat pada Tabel 3.25.



Gambar 3.29. Halaman Baca Artikel

Tabel 3.25. Spesifikasi Atribut Antarmuka Halaman Detail Artikel

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan / Keluaran
1	imageCardDetail	ImageView	Tempat yang menampung gambar yang mewakili kategori	image
2	titleDetail	TextView	Menampilkan judul dari artikel	Text
3	contentDetail	TextView	Menampilkan isi keseluruhan dari artikel	Text

BAB IV IMPLEMENTASI

Bab ini membahas implementasi dari perancangan sistem sesuai dengan perancangan yang telah dibuat. Bahasa pemrograman yang digunakan untuk implementasi sistem adalah bahasa pemrograman Java.

4.1. Lingkungan Implementasi

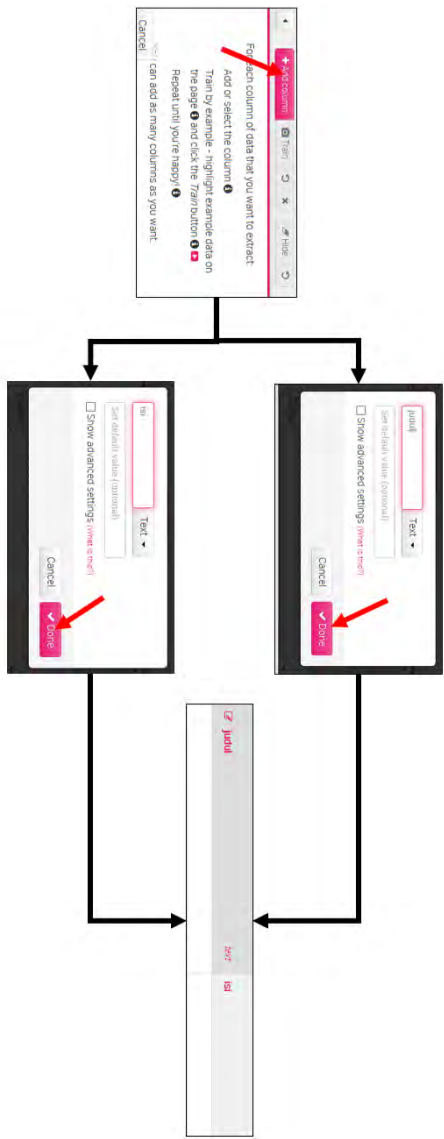
Lingkungan implementasi sistem yang digunakan untuk mengembangkan tugas akhir memiliki spesifikasi perangkat keras dan perangkat lunak seperti yang ditampilkan pada Tabel 4.1.

Tabel 4.1. Lingkungan Implementasi Sistem

Perangkat	Spesifikasi
Perangkat keras	Prosesor: Intel® Core™ i3-3110M CPU @ 2.40GHz (4 CPUs) , ~2.4GHz Memori: 4096 MB + 2048 MB
Perangkat lunak	Sistem Operasi: Microsoft Windows Embedded 8.1 Pro 64-bit Perangkat Pengembang: IDE Netbeans 8.1 dan Android Studio 1.4 Perangkat Pembantu: Notepad++, Microsoft Excel 2016, Microsoft Word 2016, Microsoft Powerpoint 2016

4.2. Implementasi Proses

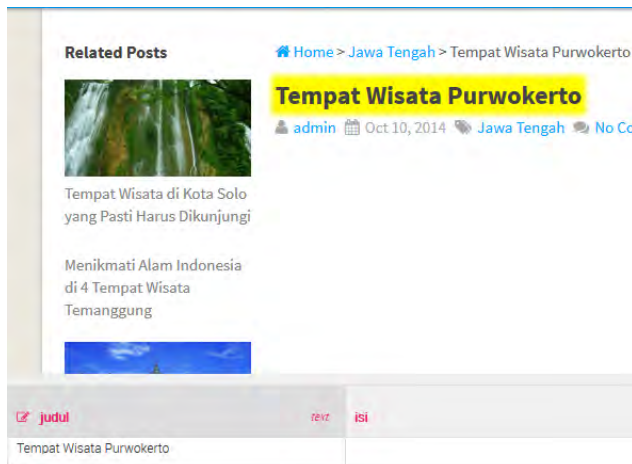
Implementasi proses dilakukan berdasarkan perancangan proses yang sudah dijelaskan pada bab analisis dan perancangan.



Gambar 4.1. Penentuan Atribut yang Akan Diambil

4.2.1. Implementasi Tahap Pengumpulan dan Persiapan Data

Subbab ini membahas implementasi tahap pengumpulan dan persiapan data yang akan diproses pada tahap selanjutnya. Implementasi tahap ini dilakukan dengan melakukan pengambilan data artikel menggunakan Import.io, parsing data JSON kedalam objek artikel, dan pembersihan *stopword*. Hasil dari tahap ini adalah sekumpulan data artikel yang sudah bersih dari *stopword* dan siap untuk diproses selanjutnya. Tahap pengumpulan dan persiapan data ini terdiri dari beberapa proses seperti yang telah dijelaskan pada subbab 3.3.1. Berikut ini adalah implementasi dari proses pengambilan data artikel menggunakan Import.io. Penentuan atribut yang akan diambil ditunjukkan pada Gambar 4.1. Pembuatan data training untuk atribut judul artikel ditunjukkan pada Gambar 4.2. Pembuatan data training untuk atribut isi artikel ditunjukkan pada Gambar 4.3. Dan pengaturan *crawler* ditunjukkan pada Gambar 4.4.



Gambar 4.2. Penentuan Elemen yang Akan Diambil Untuk Atribut Judul

Tempat Wisata Purwokerto

admin Oct 10, 2014 Jawa Tengah No Comments

Tempat Wisata Purwokerto. Berkunjung ke [Tempat wisata Purwokerto](#) merupakan suatu momen yang sangat menyenangkan. Mungkin diantara anda ada yang kampungnya terletak di Purwokerto, Jawa Tengah. Selain itu, kota ini juga memiliki berbagai macam objek wisata yang sangat memukau. Memang, objek wisata di kota Purwokerto tidak terlalu terkenal seperti [tempat wisata](#) yang ada di kota lainnya. Namun, kota ini memiliki banyak sekali objek [wisata alam](#) yang sangat indah untuk anda nikmati.

Semua objek wisata ini memiliki pemandangan yang sangat indah dan juga suasana yang sangat sejuk sehingga dapat membuat para wisatawan menjadi betah untuk terus berada di semua objek wisata yang ada di kota Purwokerto. Untuk anda yang saat ini sedang mempunyai planning untuk berkunjung ke kota Purwokerto, maka saya akan berikan beberapa tempat wisata favorit yang ada di Purwokerto, silakan disimak ya.

Isi

text

Tempat Wisata Purwokerto. Berkunjung ke Tempat wis...

Gambar 4.3. Penentuan Elemen yang Akan Diambil Untuk Atribut Isi

Where to start?

One page URL per line

```
http://wisataindonesia.co.id/tempat-wisata-purwokerto
http://wisataindonesia.co.id/mengapa-yogyakarta-begitu-unik
http://wisataindonesia.co.id/tempat-wisata-di-solo
http://wisataindonesia.co.id/lawang-sewu-semarang
http://wisataindonesia.co.id/wisata-sungai-tungtung
```

Page depth 10

Save stream (Optional)



Choose

Stream type JSON

Go

Gambar 4.4. Tampilan Konfigurasi Crawler

Setelah didapatkan hasil *crawling* dari Import.io yang berbentuk *file* JSON, proses selanjutnya adalah melakukan *parsing data* JSON kedalam objek artikel. Kelas artikel ditunjukkan pada Kode Sumber 4.1. Kelas data yang berisi kumpulan artikel ditunjukkan pada Kode Sumber 4.2. *Adapter* untuk objek artikel dan data masing–masing ditunjukkan pada Kode Sumber 4.3 dan Kode Sumber 4.4. Proses utama dalam mendapatkan objek artikel dari file JSON ditunjukkan pada Kode Sumber 4.5

```
public class Article {  
    private String[] titles;  
    private String[] contents;  
  
    public String[] getTitles() {  
        return titles;  
    }  
  
    public void setTitles(String[] titles) {  
        this.titles = titles;  
    }  
  
    public String[] getContents() {  
        return contents;  
    }  
  
    public void setContents(String[] contents) {  
        this.contents = contents;  
    }  
  
    public String toString() {  
        ...  
    }  
}
```

Kode Sumber 4.1. Kode Untuk Kelas Artikel

```

public class Data {
    private Article[] articles;

    public Article[] getArticles() {
        return articles;
    }

    public void setArticles(Article[] articles) {
        this.articles = articles;
    }

    public String toString() {
        ...
    }
}

```

Kode Sumber 4.2. Kode Untuk Kelas Data

```

...
final Article article = new Article();

if ((jsonObject.get("judul") != null) &&
    (jsonObject.get("isi") != null)) {
    final JSONArray jsonArticleTitles =
        jsonObject.get("judul").getAsJSONArray();
    final String[] titles = new
        String[jsonArticleTitles.size()];
    for (int i = 0; i < titles.length; i++) {
        final JsonElement jsonTitle =
            jsonArticleTitles.get(i);
        titles[i] = jsonTitle.getAsString();
    }

    article.setTitles(titles);

    final JSONArray jsonArticleContents =
        jsonObject.get("isi").getAsJSONArray();
    final String[] contents = new
        String[jsonArticleContents.size()];
    for (int j = 0; j < contents.length; j++) {
        final JsonElement jsonContent =
            jsonArticleContents.get(j);
        contents[j] = jsonContent.getAsString();
    }
    article.setContents(contents);
}
...

```

Kode Sumber 4.3. Kode Deserializer Artikel

```

...
final JsonObject jsonObject = json.getAsJsonObject();

final Article[] articles =
    context.deserialize(jsonObject.get("data"),
        Article[].class);

final Data data = new Data();
data.setArticles(articles);
...

```

Kode Sumber 4.4. Kode Deserializer Data

```

...
final GsonBuilder gsonBuilder = new GsonBuilder();
gsonBuilder.registerTypeAdapter(Data.class, new
    DataDeserializer());
gsonBuilder.registerTypeAdapter(Article.class, new
    ArticleDeserializer());
final Gson gson = gsonBuilder.create();
JsonReader reader = new JsonReader(new FileReader(...));

final Data data = gson.fromJson(reader, Data.class);
Article[] articles = data.getArticles();
...

```

Kode Sumber 4.5. Kode yang Menunjukkan Proses Utama Pengambilan Data JSON

Setelah data artikel sudah berhasil diambil dari file JSON, proses selanjutnya adalah pembuatan array yang berisi *stopword* seperti yang ditunjukkan oleh Kode Sumber 4.6. Implementasi proses penghapusan *stopword* pada artikel ditunjukkan oleh Kode Sumber 4.7.

```

...
List<String> stopWord = Arrays.asList(...);
...

```

Kode Sumber 4.6. Kode Untuk Pembuatan Array yang Berisi Stopword

```

...
while(rs.next()){
    String documentContent = rs.getString("isi");
    int documentId = rs.getInt("id");
    String hasil = " ";

```

```

StringTokenizer contentTokenizer = new
StringTokenizer(documentContent, ...);
while (contentTokenizer.hasMoreTokens()) {
    String tmpContent = contentTokenizer.nextToken();
    if(stopWord.contains(tmpContent.toLowerCase())){
        continue;
    }
    else{
        hasil += tmpContent + " ";
    }
}
stmt2.execute("UPDATE artikel SET bersih='" + hasil +
" ' WHERE id=" + documentId);
}
...

```

Kode Sumber 4.7. Kode Fungsi Penghapusan Stopword di dalam Artikel

4.2.2. Implementasi Tahap Perancangan Proses Pembobotan Dokumen dengan Pendekatan Fuzzy

Subbab ini membahas implementasi tahap perancangan algoritma *fuzzy* yang bertujuan untuk menghasilkan *membership function* dan *fuzzy rule* sebagai hasil akhir dari tahap ini. Sebagai data masukan yang diolah pada tahap ini adalah data artikel yang didapatkan dari tahap pengumpulan dan persiapan data. Data artikel akan dianalisa untuk menentukan *membership function* dari parameter *term frequency*, *inverse document frequency*, dan *inverse document length*. Analisa ini dilakukan dengan cara melihat jangkauan nilai yang didapatkan oleh masing – masing parameter pada dataset artikel yang ada. Proses analisa pada parameter *term frequency* ditunjukkan oleh Kode Sumber 4.8, parameter *inverse document frequency* ditunjukkan oleh Kode Sumber 4.9, parameter *inverse document length* ditunjukkan oleh Kode Sumber 4.10, dan fungsi–fungsi untuk perhitungan parameter diatas ditunjukkan pada Kode Sumber 4.11.

```

...
rs2 = stmt2.executeQuery("SELECT bersih FROM artikel ORDER
BY id ASC");
String sampelTerm = reader.nextLine();
List<Double> tfArray = new ArrayList<>();

```



```

while(rs2.next()){
    List<String> listArtikel = new ArrayList<>();
    String content = rs2.getString("bersih");
    StringTokenizer contentTokenizer = new
        StringTokenizer(content, " ");
    while(contentTokenizer.hasMoreTokens()){
        listArtikel.add(contentTokenizer.nextToken());
    }
    tfArray.add(calculator.tf(listArtikel, sampelTerm));
}
System.out.println("Nilai Tertinggi = " + max(tfArray));
System.out.println("Nilai Terendah = " + min(tfArray));
System.out.println("Nilai Rata - Rata = " +
    average(tfArray));
System.out.println("Kuartil Atas = " +
    (max(tfArray)+average(tfArray))/2);
System.out.println("Kuartil Bawah = " +
    (min(tfArray)+average(tfArray))/2);
...

```

Kode Sumber 4.8. Kode Untuk Analisa Parameter Term Frequency

```

...
List<String> sampelTermCollection = new ArrayList<>(...);
List<Double> idfArray = new ArrayList<>();
while (i < sampelTermCollection.size()){
    String temp = sampelTermCollection.get(i);
    double idf = calculator.idf(articleCollection, temp);
    idfArray.add(idf);
    i++;
}
System.out.println("Nilai Tertinggi = " + max(idfArray));
System.out.println("Nilai Terendah = " + min(idfArray));
System.out.println("Nilai Rata - Rata = " +
    average(idfArray));
System.out.println("Kuartil Atas = " +
    (max(idfArray)+average(idfArray))/2);
System.out.println("Kuartil Bawah = " +
    (min(idfArray)+average(idfArray))/2);
...

```

Kode Sumber 4.9. Kode Untuk Analisa Parameter Inverse Document Frequency

```

...
List<Double> nArray = new ArrayList<>();
while (i < articleCollection.size()){
    String temp = articleCollection.get(i);

```

```

        double n = calculator.normDocument(temp);
        nArray.add(n);
        i++;
    }
    System.out.println("Nilai Tertinggi = " + max(nArray));
    System.out.println("Nilai Terendah = " + min(nArray));
    System.out.println("Nilai Rata - Rata = " +
        average(nArray));
    System.out.println("Kuartil Atas = " +
        (max(nArray)+average(nArray))/2);
    System.out.println("Kuartil Bawah = " +
        (min(nArray)+average(nArray))/2);
    ...

```

Kode Sumber 4.10. Kode Untuk Analisa Parameter Inverse Document Length

```

...
public double tf(List<String> doc, String term) {
    double result = 0;
    for (String word : doc) {
        if (term.equalsIgnoreCase(word)) {
            result++;
        }
    }
    return result / doc.size();
}

public double idf(List<List<String>> docs, String term) {
    double n = 0;
    for (List<String> doc : docs) {
        for (String word : doc) {
            if (term.equalsIgnoreCase(word)) {
                n++;
                break;
            }
        }
    }
    if (n != 0) {
        return Math.log10(docs.size() / n);
    } else {
        return 2.8;
    }
}

public double documentNorm(List<String> doc){
    return 1/Math.sqrt(doc.size());
}...

```

Kode Sumber 4.11.Kode Untuk Perhitungan Parameter TF, IDF, dan Norm

Potongan kode diatas akan menghasilkan jangkauan nilai dari setiap parameter. Jangkauan yang dimaksud meliputi nilai tertinggi, terendah, rata-rata, kuartil atas dan kuartil bawah. Hasil analisa akan digunakan untuk mengimplementasikan *membership function* setiap parameter seperti yang ditunjukkan pada Kode Sumber 4.12 untuk parameter *term frequency* didalam dokumen, Kode Sumber 4.13 untuk parameter *term frequency* didalam *query*, Kode Sumber 4.14 untuk parameter *inverse document frequency*, Kode Sumber 4.15 untuk parameter *inverse document length*, dan Kode Sumber 4.16 untuk parameter *inverse query length*.

```
...
public double veryLow(double tf){
    if(tf >= 0.01)
    {
        return 0;
    }
    else if(tf > 0 && tf < 0.01){
        return (0.01 - tf)/0.01;
    }
    else{
        return 1;
    }
}

public double low(double tf){
    if(tf <= 0.005 || tf >= 0.02){
        return 0;
    }
    else if(tf > 0.005 && tf <= 0.0125){
        return (tf-0.005)/0.0075;
    }
    else if(tf >= 0.0125 && tf <0.02){
        return (0.2-tf)/0.0075;
    }
    else{
        return 1;
    }
}

public double medium(double tf){
    if(tf <= 0.015 || tf >= 0.075){
        return 0;
    }
    else if(tf > 0.015 && tf <= 0.045){
```

```

        return (tf-0.015)/0.03;
    }
    else if(tf >= 0.045 && tf < 0.075){
        return (0.075-tf)/0.03;
    }
    else{
        return 1;
    }
}

public double high(double tf){
    if(tf <= 0.06 || tf >= 0.12){
        return 0;
    }
    else if(tf > 0.06 && tf <= 0.09){
        return (tf-0.06)/0.03;
    }
    else if(tf >= 0.09 && tf < 0.12){
        return (1.2-tf)/0.03;
    }
    else{
        return 1;
    }
}

public double veryHigh(double tf){
    if(tf <= 0.1){
        return 0;
    }
    else if(tf > 0.1 && tf < 1){
        return (tf-0.1)/0.9;
    }
    else if(tf >= 1){
        return 1;
    }
    else{
        return 1;
    }
}
...

```

Kode Sumber 4.12. Kode Untuk Membership Function Term Frequency di dalam Dokumen

```

...
public double veryLow(double tf){
    if(tf >= 0.15)
    {
        return 0;
    }
}

```

```

    }
    else if(tf > 0 && tf < 0.15){
        return (0.15 - tf)/0.15;
    }
    else{
        return 1;
    }
}

public double low(double tf){
    if(tf <= 0.1 || tf >= 0.3){
        return 0;
    }
    else if(tf > 0.1 && tf <= 0.2){
        return (tf-0.1)/0.1;
    }
    else if(tf >= 0.2 && tf <0.3){
        return (0.3-tf)/0.1;
    }
    else{
        return 1;
    }
}

public double medium(double tf){
    if(tf <= 0.25 || tf >= 0.45){
        return 0;
    }
    else if(tf > 0.25 && tf <= 0.35){
        return (tf-0.25)/0.1;
    }
    else if(tf >= 0.35 && tf < 0.45){
        return (0.45-tf)/0.1;
    }
    else{
        return 1;
    }
}

public double high(double tf){
    if(tf <= 0.4 || tf >= 0.6){
        return 0;
    }
    else if(tf > 0.4 && tf <= 0.5){
        return (tf-0.4)/0.1;
    }
    else if(tf >= 0.5 && tf < 0.6){
        return (0.6-tf)/0.1;
    }
    else{
        return 1;
    }
}

```

```

}
public double veryHigh(double tf){
    if(tf <= 0.55){
        return 0;
    }
    else if(tf > 0.55 && tf < 1){
        return (tf-0.55)/0.45;
    }
    else if(tf >= 1){
        return 1;
    }
    else{
        return 1;
    }
}
...

```

Kode Sumber 4.13. Kode Untuk Membership Function Term Frequency di dalam Query

```

...
public double veryLow(double idf){
    if(idf >= 0.7)
    {
        return 0;
    }
    else if(idf > 0 && idf < 0.7){
        return (0.7-idf)/0.7;
    }
    else{
        return 1;
    }
}
public double low(double idf){
    if(idf <= 0.4 || idf >= 1.1){
        return 0;
    }
    else if(idf > 0.4 && idf <= 0.75){
        return (idf-0.4)/0.35;
    }
    else if(idf >= 0.75 && idf < 1.1){
        return (1.1-idf)/0.35;
    }
    else{
        return 1;
    }
}
public double medium(double idf){
    if(idf <= 0.9 || idf >= 1.7){

```

```

        return 0;
    }
    else if(idf > 0.9 && idf <= 1.3){
        return (idf-0.9)/0.4;
    }
    else if(idf >= 1.3 && idf < 1.7){
        return (1.7-idf)/0.4;
    }
    else{
        return 1;
    }
}
public double high(double idf){
    if(idf <= 1.4 || idf >= 2.4){
        return 0;
    }
    else if(idf > 1.4 && idf <= 1.9){
        return (idf-1.4)/0.5;
    }
    else if(idf >= 1.9 && idf < 2.4){
        return (2.4-idf)/0.5;
    }
    else{
        return 1;
    }
}
public double veryHigh(double idf){
    if(idf <= 2){
        return 0;
    }
    else if(idf > 2 && idf < 2.8){
        return (idf-2)/0.8;
    }
    else if(idf >= 2.8){
        return 1;
    }
    else{
        return 1;
    }
}
...

```

**Kode Sumber 4.14. Kode Untuk Membership Function
Inverse Document Frequency**

```

...
public double veryLow(double n){
    if(n >= 0.04)
    {

```

```

        return 0;
    }
    else if(n > 0 && n < 0.04){
        return (0.04 - n)/0.04;
    }
    else{
        return 1;
    }
}

public double low(double n){
    if(n <= 0.03 || n >= 0.06){
        return 0;
    }
    else if(n > 0.03 && n <= 0.045){
        return (n-0.03)/0.015;
    }
    else if(n >= 0.045 && n <0.06){
        return (0.06-n)/0.015;
    }
    else{
        return 1;
    }
}

public double medium(double n){
    if(n <= 0.05 || n >= 0.09){
        return 0;
    }
    else if(n > 0.05 && n <= 0.07){
        return (n-0.05)/0.02;
    }
    else if(n >= 0.07 && n < 0.09){
        return (0.09-n)/0.02;
    }
    else{
        return 1;
    }
}

public double high(double n){
    if(n <= 0.08 || n >= 0.12){
        return 0;
    }
    else if(n > 0.08 && n <= 0.1){
        return (n-0.08)/0.02;
    }
    else if(n >= 0.1 && n < 0.12){
        return (0.12-n)/0.02;
    }
    else{
        return 1;
    }
}

```



```

    }
}
public double veryHigh(double n){
    if(n <= 0.11){
        return 0;
    }
    else if(n > 0.11 && n < 1){
        return (n-0.11)/0.89;
    }
    else if(n >= 1){
        return 1;
    }
    else{
        return 1;
    }
}
...

```

Kode Sumber 4.15. Kode Untuk Membership Function Inverse Document Length

```

...
public double veryLow(double n){
    if(n >= 0.1)
    {
        return 0;
    }
    else if(n > 0 && n < 0.1){
        return (0.1 - n)/0.1;
    }
    else{
        return 1;
    }
}
public double low(double n){
    if(n <= 0.05 || n >= 0.25){
        return 0;
    }
    else if(n > 0.05 && n <= 0.15){
        return (n-0.05)/0.1;
    }
    else if(n >= 0.15 && n < 0.25){
        return (0.25-n)/0.1;
    }
    else{
        return 1;
    }
}
public double medium(double n){

```

```

        if(n <= 0.2 || n >= 0.4){
            return 0;
        }
        else if(n > 0.2 && n <= 0.3){
            return (n-0.2)/0.1;
        }
        else if(n >= 0.3 && n < 0.4){
            return (0.4-n)/0.1;
        }
        else{
            return 1;
        }
    }
}

public double high(double n){
    if(n <= 0.3 || n >= 0.7){
        return 0;
    }
    else if(n > 0.3 && n <= 0.5){
        return (n-0.3)/0.2;
    }
    else if(n >= 0.5 && n < 0.7){
        return (0.7-n)/0.2;
    }
    else{
        return 1;
    }
}

public double veryHigh(double n){
    if(n <= 0.5){
        return 0;
    }
    else if(n > 0.5 && n < 1){
        return (n-0.5)/0.5;
    }
    else if(n >= 1){
        return 1;
    }
    else{
        return 1;
    }
}

...

```

**Kode Sumber 4.16. Kode Untuk Membership Function
Inverse Query Length**

Potongan-potongan kode di atas adalah implementasi *membership function* yang merupakan elemen dari logika *fuzzy*.

Elemen lain dalam logika *fuzzy* adalah *fuzzy rule* yang implementasinya ditunjukkan pada Kode Sumber 4.17.

```
...
public List<Double> calculate(List<Double> tf,
List<Double> idf, List<Double> n){
    List<Double> result = new ArrayList<>();
    result.add(Math.min(Math.max(tf.get(0), tf.get(1)),
Math.min(Math.max(idf.get(0), idf.get(1)),
Math.max(n.get(0), n.get(1)))));
    result.add(Math.min(Math.max(tf.get(0), tf.get(1)),
Math.min(Math.max(idf.get(0), idf.get(1)), n.get(2)))));
    result.add(Math.min(Math.max(tf.get(0), tf.get(1)),
Math.min(Math.max(idf.get(0), idf.get(1)),
Math.max(n.get(3), n.get(4)))));
    ...
    result.add(Math.min(Math.max(tf.get(3), tf.get(4)),
Math.min(Math.max(idf.get(3), idf.get(4)),
Math.max(n.get(0), n.get(1)))));
    result.add(Math.min(Math.max(tf.get(3), tf.get(4)),
Math.min(Math.max(idf.get(3), idf.get(4)), n.get(2)))));
    result.add(Math.min(Math.max(tf.get(3), tf.get(4)),
Math.min(Math.max(idf.get(3), idf.get(4)),
Math.max(n.get(3), n.get(4)))));
    return result;
}
...
```

Kode Sumber 4.17. Kode Untuk Implementasi Fuzzy Rule

4.2.3. Implementasi Tahap Perancangan Web Service

Subbab ini membahas implementasi tahap perancangan *web service*. Hasil dari implementasi tahap perancangan proses pembobotan dokumen dengan pendekatan *fuzzy* yang berupa *membership function* dan *fuzzy rule* akan diletakkan dalam *web service*. Di dalam tahap perancangan *web service* ini terdapat beberapa proses yang terlibat. Dimulai dari proses pengambilan artikel serta proses tokenisasi artikel seperti yang ditunjukkan oleh Kode Sumber 4.18 dan tokenisasi *query* masukan seperti yang ditunjukkan pada Kode Sumber 4.19.

```
...
while(rs.next()){
    String documentTitle = rs.getString("judul");
    String documentContent = rs.getString("isi");
```

```

String documentCategory = rs.getString("kategori");
titleCollection.add(documentTitle);
contentCollection.add(documentContent);
categoryCollection.add(documentCategory);
documentToken = new ArrayList<>();
title = new ArrayList<>();

StringTokenizer contentTokenizer= new StringTokenizer(
    documentContent, " !@#$%^&*()-_+=|;:'.\"/<>?");
while (contentTokenizer.hasMoreTokens()) {
    String tmpContent = contentTokenizer.nextToken();
    if(stopWord.contains(tmpContent.toLowerCase())){
        continue;
    }
    documentToken.add(tmpContent.toLowerCase());
}
StringTokenizer titleTokenizer = new StringTokenizer(
    documentTitle, " !@#$%^&*()-_+=|;:'.\"/<>?");
while (titleTokenizer.hasMoreTokens()) {
    String tmpTitle = titleTokenizer.nextToken();
    if(stopWord.contains(tmpTitle.toLowerCase())){
        continue;
    }
    title.add(tmpTitle.toLowerCase());
}
documents.add(documentToken);
titles.add(title);
}
...

```

Kode Sumber 4.18. Kode Untuk Mengambil Artikel dan Tokenisasi

```

...
String inputUser = query;
ArrayList<String> query = new ArrayList<>();
StringTokenizer queryTokenizer = new
StringTokenizer(inputUser, " \!@#$%^&*()-_+=|;:'.\"/<>?");
while (queryTokenizer.hasMoreTokens()) {
    query.add(queryTokenizer.nextToken());
}
...

```

Kode Sumber 4.19. Kode Untuk Mengambil Query dan Tokenisasi

Dari potongan kode diatas akan didapatkan hasil tokenisasi artikel yang disimpan dalam variabel *documents* untuk

hasil tokenisasi isi artikel dan variabel *titles* untuk hasil tokenisasi judul artikel. Hasil tokenisasi *query* disimpan dalam variabel *query*. Proses selanjutnya setelah mendapatkan hasil tokenisasi adalah melakukan pembobotan setiap dokumen terhadap *query* seperti yang ditunjukkan pada Kode Sumber 4.20.

```
...
ArrayList<String> scoreCollection = new ArrayList();
int i = 0;
ModifiedFuzzyCalculator calculator = new
    ModifiedFuzzyCalculator();
while (i < documents.size()) {
    int j = 0;
    double contentValue = 0;
    double titleValue = 0;
    while (j < query.size()) {
        contentValue += calculator.calculate(documents.get(i),
            documents, query.get(j), query);
        titleValue += calculator.calculate(titles.get(i),
            titles, query.get(j), query);
        j++;
    }
    contentValue = contentValue / query.size();
    titleValue = titleValue / query.size();
    tfIdfValue = ((5 * tfIdfValue) + (5 * titleValue))/10;
    String tfIdfValueString = String.valueOf(tfIdfValue);
    scoreCollection.add(tfIdfValueString);
    i++;
}
...
```

Kode Sumber 4.20. Kode Untuk Melakukan Pembobotan Dokumen Terhadap Query

Potongan kode diatas akan menghasilkan sekumpulan nilai hasil pembobotan dokumen terhadap *query* yang disimpan di dalam variabel *scoreCollection*. Proses selanjutnya yang harus dilakukan adalah membuat objek artikel yang berisi elemen-elemen dari artikel seperti yang ditunjukkan pada Kode Sumber 4.21 .

```
...
ArrayList<TfIdfWithArticleIndex> arrayToSort = new
    ArrayList<TfIdfWithArticleIndex>();
```

```

i = 0;
while (i < tfIdf.size()) {
    TfIdfWithArticleIndex tempOfTfIdf = new
        TfIdfWithArticleIndex(i, tfIdf.get(i),
            titleCollection.get(i), contentCollection.get(i),
            categoryCollection.get(i));
    arrayToSort.add(tempOfTfIdf);
    i++;
}
...

```

Kode Sumber 4.21. Kode Untuk Pembuatan Objek Artikel

Potongan kode diatas menghasilkan objek – objek artikel berupa *TfIdfWithArticleIndex* yang disimpan dalam variabel *arrayToSort*. Proses selanjutnya adalah mengurutkan objek–objek yang ada dalam variable *arrayToSort* berdasarkan nilai hasil pembobotan dari objek artikel. Implementasi proses pengurutan ini ditunjukkan pada Kode Sumber 4.22.

```

...
i = 0;
while (i < arrayToSort.size()) {
    int j = i + 1;
    while (j < arrayToSort.size()) {
        if(arrayToSort.get(i).compareTo(arrayToSort.get(j))
            < 0) {
            TfIdfWithArticleIndex tempOfTfIdf =
                arrayToSort.get(i);
            arrayToSort.set(i, arrayToSort.get(j));
            arrayToSort.set(j, tempOfTfIdf);
        }
        j++;
    }
    i++;
}
...

```

Kode Sumber 4.22. Kode Untuk Pengurutan Artikel Berdasarkan Nilai Hasil Pembobotan

Potongan kode diatas akan menghasilkan objek–objek artikel yang ada dalam variabel *arrayToSort* terurut berdasarkan nilai hasil pembobotannya. Proses selanjutnya adalah menentukan jumlah artikel yang ingin dikembalikan dan mengembalikan hasilnya seperti yang ditunjukkan pada Kode Sumber 4.23.

```

...
i = 0;
List<TfIdfWithArticleIndex> searchResult = new
    ArrayList<>();
while (i < tfIdf.size()) {
    if (arrayToSort.get(i).getArticleTfIdf().equals("0.0"))
    {
        break;
    }
    else if(i == 10){
        break;
    }
    else {
        searchResult.add(arrayToSort.get(i));
    }
    i++;
}
...

```

Kode Sumber 4.23. Kode Untuk Menentukan Jumlah Artikel yang Ingin Dikembalikan

Potongan–potongan kode diatas menunjukkan keseluruhan proses berjalannya *web service*. Karena *return value* dari *web service* berupa sekumpulan objek artikel yang merupakan *custom class*, maka perlu ditambahkan sebuah definisi agar objek ini dapat dikenali sebagai *response* dari *web service* seperti yang ditunjukkan pada Kode Sumber 4.24.

```

...
@XmlAccessorType(XmlAccessType.FIELD)
@XmlType(name = "TfIdfWithArticleIndex")
public class TfIdfWithArticleIndex {
    @XmlElement(name = "articleIndex")
    private int articleIndex;
    @XmlElement(name = "articleTfIdf")
    private String articleTfIdf;
    @XmlElement(name = "articleTitle")
    private String articleTitle;
    @XmlElement(name = "articleContent")
    private String articleContent;
    @XmlElement(name = "articleCategory")
    private String articleCategory;
}
...

```

Kode Sumber 4.24. Kode Untuk Menentukan Skema Response dari Web Service

4.2.4. Implementasi Tahap Perancangan Aplikasi Mobile

Subbab ini membahas implementasi tahap perancangan aplikasi *mobile*. Aplikasi *mobile* ini akan digunakan untuk berkomunikasi dengan *web service* yang diimplementasikan pada tahap sebelumnya. Proses pertama adalah menentukan atribut–atribut yang dibutuhkan untuk koneksi ke *web service* seperti yang ditunjukkan pada Kode Sumber 4.25.

```
...
String NAMESPACE = "http://service.fuzzy.com/";
String METHOD_NAME = "doSearch";
String SOAP_ACTION = "http://service.fuzzy.com/doSearch";
String URL =
    "http://10.0.2.2:8080/FuzzyWebService/FuzzyService";
...
```

Kode Sumber 4.25. Kode Untuk Menentukan Atribut–Atribut Koneksi ke Web Service

Proses selanjutnya yang dilakukan setelah menentukan atribut–atribut koneksi ke *web service* adalah menentukan parameter – parameter masukan yang dibutuhkan oleh *web service*. Proses penentuan parameter ini ditunjukkan pada Kode Sumber 4.26.

```
...
SoapObject request = new SoapObject(NAMESPACE,
    METHOD_NAME);
PropertyInfo p = new PropertyInfo();
p.setName("query");
p.setValue(params[0]);
p.setType(String.class);
request.addProperty(p);
...
```

Kode Sumber 4.26. Kode Untuk Menentukan Parameter Masukan Web Service

Setelah menentukan parameter–parameter yang dibutuhkan oleh *web service* maka komunikasi ke *web service* dilakukan dengan menggunakan *HttpTransport*. Proses ini ditunjukkan pada Kode Sumber 4.27.


```

...
SoapSerializationEnvelope envelope = new
SoapSerializationEnvelope(SoapEnvelope.VER11);

envelope.setOutputSoapObject(request);
HttpTransportSE androidHttpTransport = new
    HttpTransportSE(URL);
List<ArticleInfo> result = new ArrayList<>();

try {
    androidHttpTransport.call(SOAP_ACTION, envelope);
    SoapObject resultRequest =
        (SoapObject)envelope.bodyIn;
...

```

Kode Sumber 4.27. Kode Untuk Melakukan Koneksi ke Web Service

Potongan kode diatas akan mengirimkan *request* ke *web service* yang kemudian akan menghasilkan *response* dari *web service*. *Response* ini berupa sekumpulan objek artikel yang atribut-atributnya sudah dijelaskan pada tahap sebelumnya. Proses pengambilan elemen-elemen yang dibutuhkan dari *response web service* ditunjukkan pada Kode Sumber 4.28.

```

...
for(int i = 0; i < resultRequest.getPropertyCount(); i++){
    Object property = resultRequest.getProperty(i);
    if(property instanceof SoapObject){
        SoapObject finalObject = (SoapObject)property;
        ArticleInfo ai = new ArticleInfo();
        ai.title =
            finalObject.getProperty("articleTitle").toString();
        ai.content =
            finalObject.getProperty("articleContent").toString();
        ai.summary = ai.content.substring(0,
            Math.min(ai.content.length(), 150));
        ai.category =
            finalObject.getProperty("articleCategory").toString();
        result.add(ai);
    }
}
...

```

Kode Sumber 4.28. Kode Untuk Melakukan Pengambilan Elemen dari Response Web Service

4.3. Implementasi Antarmuka Pengguna

Implementasi tampilan antarmuka pengguna pada perangkat lunak berbasis *mobile* dan berjalan pada sistem operasi Android dilakukan dengan menggunakan berkas XML. Berikut ini dijelaskan mengenai implementasi tampilan antarmuka pengguna yang terdapat pada perangkat lunak.

4.3.1. Slide Drawer Navigasi

Slide drawer navigasi ini merupakan elemen utama yang ada pada setiap halaman dan digunakan untuk melakukan navigasi antar halaman di dalam aplikasi. Rancangan mengenai elemen ini dapat dilihat pada subbab 3.4.1 dan implementasinya ditunjukkan pada Kode Sumber 4.29.

```
<android.support.v4.widget.DrawerLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/drawerLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <!-- The main content view -->
    <RelativeLayout
        android:id="@+id/mainContent"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <!-- The navigation drawer -->
    <RelativeLayout
        android:layout_width="280dp"
        android:layout_height="match_parent"
        android:id="@+id/drawerPane"
        android:layout_gravity="start">

        <!-- List of Actions (pages) -->
        <ListView
            android:id="@+id/navList"
            android:layout_width="280dp"
            android:layout_height="match_parent"
            android:choiceMode="singleChoice"
            android:background="#ffffffff" />
```

```

        </RelativeLayout>

    </android.support.v4.widget.DrawerLayout>

```

Kode Sumber 4.29. Kode Implementasi Slide Drawer Navigasi

Dari potongan kode diatas ditunjukkan implementasi *slide drawer* menggunakan *widget DrawerLayout* yang didalamnya terdapat sebuah *list* yang berisi pilihan navigasi.

4.3.2. Halaman Utama

Halaman ini adalah halaman pertama yang ditemui oleh pengguna ketika membuka aplikasi ini. Rancangan mengenai halaman ini dapat dilihat pada subbab 3.4.2 dan implementasinya ditunjukkan pada Kode Sumber 4.30

```

<FrameLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context="com.looper.loop.PreferencesFragment">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/homeCardList"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        />

</FrameLayout>

```

Kode Sumber 4.30. Kode Implementasi Halaman Utama

Dari potongan kode diatas ditunjukkan terdapat satu buah elemen di dalam halaman utama. Elemen ini adalah *recyclerView* yang digunakan untuk menampilkan beberapa artikel acak.

4.3.3. Halaman Pemilihan Kategori

Halaman ini adalah halaman yang digunakan oleh pengguna untuk memilih kategori artikel yang diinginkan. Halaman ini terdiri dari 2 tahap, yang pertama adalah tahap untuk melakukan pemilihan kategori. Tahap yang kedua adalah menampilkan daftar artikel sesuai kategori yang dipilih. Rancangan mengenai halaman ini dapat dilihat pada subbab 3.4.3 dan implementasinya bisa dilihat pada Kode Sumber 4.31 untuk halaman pemilihan kategori dan Kode Sumber 4.32 untuk halaman menampilkan artikel sesuai kategori.

```
<FrameLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.looper.loop.PreferencesFragment">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <Button
            android:id="@+id/buttonGunung"
            ...
        />
        <Button
            android:id="@+id/buttonPantai"
            ...
        />
        <Button
            android:id="@+id/buttonCandi"
            ...
        />
        <Button
            android:id="@+id/buttonHotel"
            ...
        />
        <Button
            android:id="@+id/buttonKuliner"
            ...
        />
    </LinearLayout>
</FrameLayout>
```

Kode Sumber 4.31. Kode Implementasi Halaman Pemilihan Kategori

```

<FrameLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.revmedia.tugasakhir.CategoryListFragmen
t">
    <android.support.v7.widget.RecyclerView
        android:id="@+id/cardListCategory"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
    />
</FrameLayout>

```

Kode Sumber 4.32. Kode Implementasi Halaman Menampilkan Artikel Berdasar Kategori

Pada Kode Sumber 4.31 ditunjukkan beberapa *button* yang digunakan untuk memilih kategori, sedangkan pada Kode Sumber 4.32 ditunjukkan sebuah elemen *RecyclerView* yang digunakan untuk menampilkan daftar artikel berdasarkan kategori yang sudah dipilih pada halaman sebelumnya.

4.3.4. Halaman Pencarian

Halaman ini merupakan halaman yang digunakan oleh pengguna untuk melakukan pencarian artikel. Rancangan mengenai halaman ini dapat dilihat pada subbab 3.4.4 dan implementasinya bisa dilihat pada Kode Sumber 4.33.

```

<FrameLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.looper.loop.PreferencesFragment">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <SearchView
            android:id="@+id/doSearch"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:queryHint="Search Here"
        />
    </LinearLayout>
</FrameLayout>

```

```

        android:iconifiedByDefault="false"
        android:onClick="searchCommited"/>

        <android.support.v7.widget.RecyclerView
            android:id="@+id/cardList"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
        />
    </LinearLayout>
</FrameLayout>

```

Kode Sumber 4.33. Kode Implementasi Halaman Pencarian Artikel

Dari potongan kode diatas ditunjukkan pada halaman pencarian terdapat dua buah elemen yaitu sebuah *SearchView* yang berfungsi untuk memasukkan *query* pencarian dan sebuah *RecyclerView* yang berfungsi untuk menampilkan hasil pencarian.

4.3.5. Halaman Detail Artikel

Halaman ini merupakan halaman yang digunakan oleh pengguna untuk melihat isi artikel. Halaman ini dijalankan ketika pengguna memilih sebuah artikel dari *RecyclerView* yang ada pada halaman–halaman sebelumnya. Rancangan mengenai halaman ini dapat dilihat pada subbab 3.4.5 dan implementasinya bisa dilihat pada Kode Sumber 4.34.

```

<FrameLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.revmedia.tugasakhir.ArticleFragment">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <RelativeLayout
            ... >

            <ImageView
                android:id="@+id/imageCardDetail"
                android:layout_width="match_parent"
                android:layout_height="140dp" />

```

```

<TextView
    android:id="@+id/titleDetail"
    android:layout_below="@+id/imageCardDetail"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textStyle="bold"/>
<TextView
    android:id="@+id/txtContentDetail"
    android:layout_below="@+id/titleDetail"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
</RelativeLayout>
</ScrollView>

```

Kode Sumber 4.34. Kode Implementasi Halaman Detail Artikel

Dari potongan kode diatas ditunjukkan terdapat beberapa elemen di dalam halaman detail artikel. Elemen pertama adalah sebuah *ImageView* yang menampilkan gambar sesuai kategori artikel. Elemen lainnya adalah dua buah *TextView* yang masing-masing menampilkan judul dan isi artikel.

[Halaman ini sengaja dikosongkan]

BAB V

UJI COBA DAN EVALUASI

Bab ini membahas uji coba dan evaluasi terhadap perangkat lunak yang telah dikembangkan dari implementasi *fuzzy logic based ranking function* dalam sistem informasi pencarian artikel pariwisata berbasis *mobile*.

5.1. Lingkungan Uji Coba

Lingkungan uji coba yang digunakan dalam pembuatan tugas akhir ini meliputi perangkat lunak dan perangkat keras yang digunakan untuk melakukan uji coba implementasi *fuzzy logic based ranking function* dalam sistem informasi pencarian artikel pariwisata berbasis *mobile*. Lingkungan uji coba merupakan sebuah perangkat *mobile* yang berupa *emulator* Android Studio dengan spesifikasi sebagai berikut:

- Emulator : Nexus 5 API 23 x86
- Resolusi : 1080 x 1920: xxhdpi
- Sistem Operasi : Android 6.0 Marshmallow
- RAM : 2 GB
- VM Heap : 512 MB
- Penyimpanan Internal : 200 MB

5.2. Data Uji Coba

Data yang digunakan untuk uji coba implementasi *fuzzy logic based ranking function* dalam sistem informasi pencarian artikel pariwisata berbasis *mobile* adalah data *query* pencarian yang berupa data teks. Data uji coba ditentukan dengan cara menentukan sebuah *query* secara acak, kemudian dilakukan pencocokan *query* dengan dataset secara manual. Jika ditemui terdapat 10 atau lebih artikel yang sesuai dengan *query*, maka *query* tersebut dapat dijadikan sebagai data uji coba. Tiap *query* uji coba memiliki panjang beragam antara 2-4 *term* pada tiap *query*.

Data *query* yang digunakan sebagai data uji coba beserta contoh judul artikel yang relevan dengan *query* tersebut ditunjukkan pada Lampiran 2.

5.3. Skenario Uji Coba

Pada subbab ini akan dijelaskan mengenai skenario uji coba yang telah dilakukan. Skenario uji coba akan dilakukan dengan menggunakan 11 *query* uji coba yang dijelaskan pada bagian Lampiran 2. Terdapat beberapa skenario uji coba yang telah dilakukan, diantaranya yaitu:

1. Perbandingan hasil akurasi pencarian artikel antara *fuzzy* yang menggunakan 3 parameter (*tf*, *idf*, dan *norm*) dan *fuzzy* yang menggunakan 2 parameter (*tf* dan *idf*).
2. Perbandingan hasil akurasi pencarian artikel antara *fuzzy* yang menggunakan seluruh *rule* dan *fuzzy* yang menggunakan *rule-rule* hasil penggabungan.
3. Perbandingan hasil akurasi pencarian artikel antara *fuzzy* yang menggunakan FLC (*fuzzy logic controller*) *query* dan *fuzzy* yang tidak menggunakan FLC *query*.
4. Perbandingan hasil akurasi pencarian artikel antara pembobotan dengan pendekatan *fuzzy* dan tanpa pendekatan *fuzzy*.

Dalam setiap skenario uji coba akan dilakukan perhitungan *precision* dan *recall* untuk menghitung tingkat performa dari pengambilan data. Perhitungan *precision* dan *recall* masing-masing ditunjukkan pada Persamaan 5.1 dan Persamaan 5.2

$$Precision = \frac{|R_a|}{|A|} \quad (5.1)$$

$$Recall = \frac{|R_a|}{|R|} \quad (5.2)$$

Tiap variabel dalam persamaan diatas dapat dijelaskan sebagai berikut $|A|$ adalah jumlah dokumen yang berhasil diambil oleh setiap metode pembobotan yang diujikan dan $|R|$ adalah jumlah seluruh dokumen yang relevan dengan masing-masing *query* uji coba, dan $|R_a|$ adalah jumlah dokumen relevan yang berhasil diambil oleh setiap metode pembobotan.

5.4. Skenario Pengujian 1: Perbandingan Penggunaan Parameter Norm Terhadap Hasil Akurasi Pencarian Artikel

Pada subbab ini akan dijelaskan perbandingan hasil akurasi pencarian artikel antara pendekatan *fuzzy* yang menggunakan 3 parameter (*tf*, *idf*, dan *norm*) dengan pendekatan *fuzzy* yang menggunakan 2 parameter (*tf* dan *idf*). Kedua pendekatan *fuzzy* ini menggunakan *membership function* yang sama. Sedangkan untuk *rule* yang digunakan, *fuzzy* 3 parameter menggunakan *rule* yang dijelaskan pada Tabel 3.14 dan *fuzzy* 2 parameter menggunakan *rule berbeda* yang dibuat berdasarkan diagram hubungan *tf* dan *idf* pada Gambar 2.5. Rule yang digunakan pada *fuzzy* 2 parameter ditunjukkan pada Tabel 5.1. Hasil perbandingan nilai rata-rata *precision* dan *recall* dari *fuzzy* yang menggunakan 3 parameter dengan yang menggunakan 2 parameter ditunjukkan pada Tabel 5.2.

Tabel 5.1. Rule Pada Fuzzy 2 Parameter

No	TF	IDF	THEN	W
1	Very Low	Very Low	THEN	Low
2	Very Low	Low	THEN	Low
3	Very Low	Medium	THEN	Low
4	Very Low	High	THEN	Medium
5	Very Low	Very High	THEN	Medium
6	Low	Very Low	THEN	Low
7	Low	Low	THEN	Low
8	Low	Medium	THEN	Low
9	Low	High	THEN	Medium

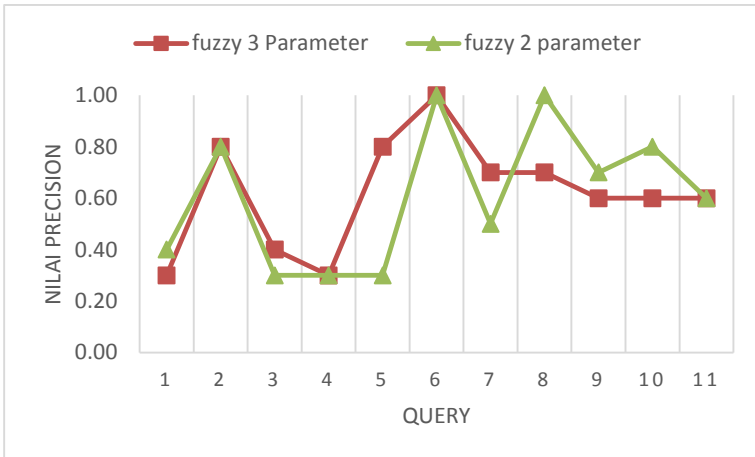
No	TF	IDF	THEN	W
10	Low	Very High	THEN	Medium
11	Medium	Very Low	THEN	Low
12	Medium	Low	THEN	Medium
13	Medium	Medium	THEN	Medium
14	Medium	High	THEN	Medium
15	Medium	Very High	THEN	Medium
16	High	Very Low	THEN	Medium
17	High	Low	THEN	Medium
18	High	Medium	THEN	Medium
19	High	High	THEN	High
20	High	Very High	THEN	High
21	Very High	Very Low	THEN	Medium
22	Very High	Low	THEN	Medium
23	Very High	Medium	THEN	Medium
24	Very High	High	THEN	High
25	Very High	Very High	THEN	High

Tabel 5.2. Perbandingan Rata-Rata Precision dan Recall Antara Fuzzy 3 Parameter dengan Fuzzy 2 Parameter

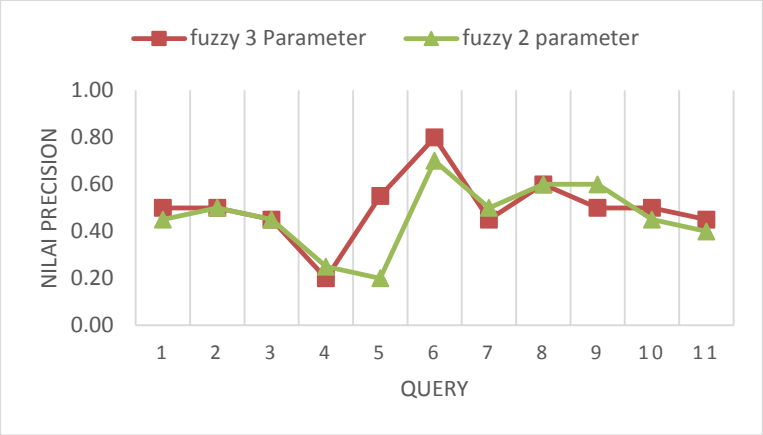
Metode Fuzzy	10 Dokumen Pertama		20 Dokumen Pertama		30 Dokumen Pertama		Time (s)
	Precision	Recall	Precision	Recall	Precision	Recall	
3 Parameter	0.6182	0.3926	0.5000	0.6361	0.3848	0.7262	1.8146
2 Parameter	0.6091	0.3966	0.4636	0.6083	0.3606	0.6975	1.6605

Dari tabel diatas dapat dilihat perbedaan *precision* dan *recall* antara pendekatan *fuzzy* yang menggunakan 3 parameter dan *fuzzy* yang menggunakan 2 parameter tidak terlalu jauh jika dilihat pada 10 dokumen pertama yang berhasil diambil. Dari 10 dokumen pertama yang berhasil diambil terdapat perbedaan nilai sebesar

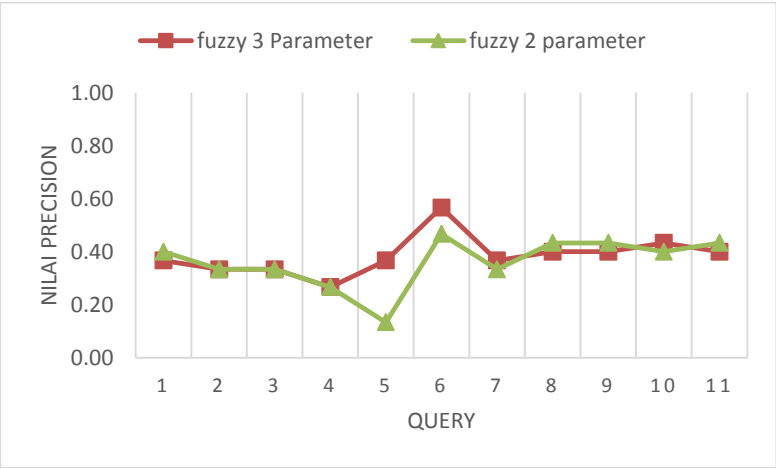
0.0091 pada *precision* dan 0.0040 pada *recall*. Perbedaan mulai terlihat jika dilihat pada 20 dokumen pertama yang berhasil diambil, terdapat perbedaan nilai sebesar 0.0364 pada *precision* dan 0.0278 pada *recall*. Perbedaan yang cukup besar juga terlihat pada 30 dokumen pertama yang berhasil diambil. Pada 30 dokumen pertama yang berhasil diambil, terdapat perbedaan nilai sebesar 0.0242 pada *precision* dan 0.0287 pada *recall*. Grafik yang menunjukkan hasil perhitungan *precision* dan *recall* pada tiap *query* ditunjukkan pada Gambar 5.1 hingga Gambar 5.6. Tiap grafik menggambarkan hubungan antara nilai *precision* atau *recall* (sumbu Y) dengan setiap *query* uji coba (sumbu X).



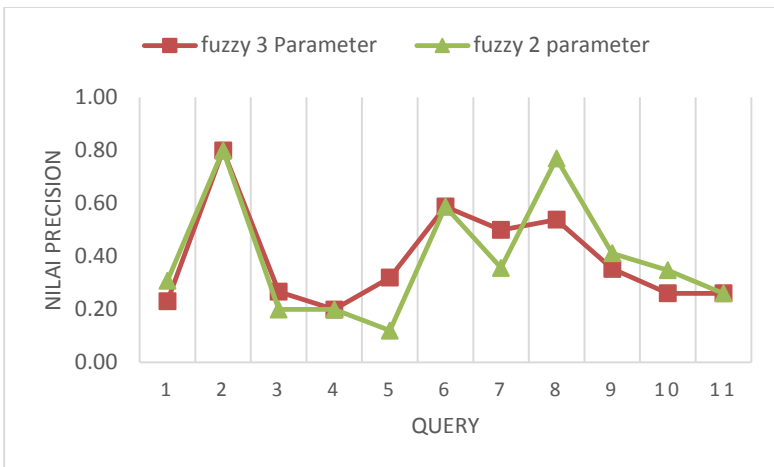
Gambar 5.1. Nilai Precision Pada 10 Dokumen Pertama Berdasarkan 11 Data Uji Coba



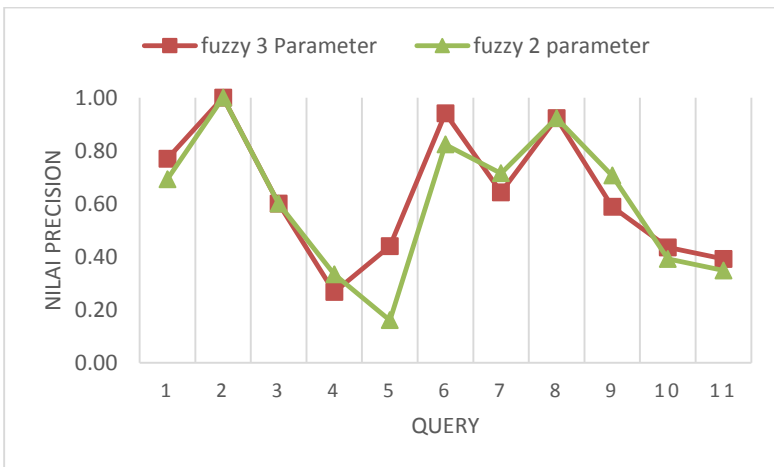
Gambar 5.2. Nilai Precision Pada 20 Dokumen Pertama Berdasarkan 11 Data Uji Coba



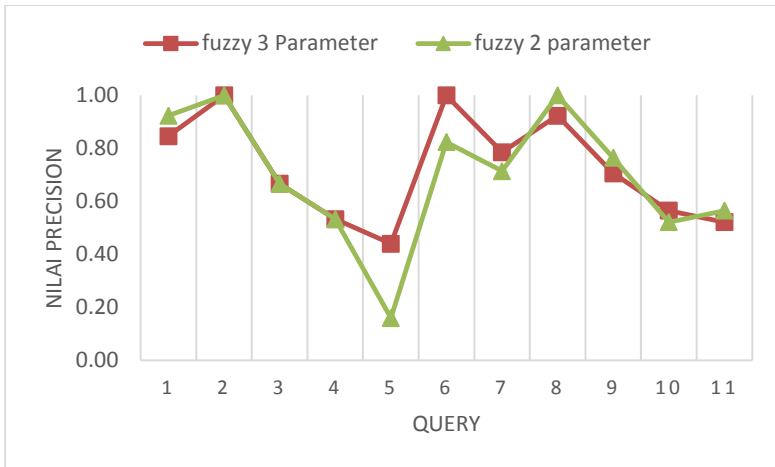
Gambar 5.3. Nilai Precision Pada 30 Dokumen Pertama Berdasarkan 11 Data Uji Coba



Gambar 5.4. Nilai Recall Pada 10 Dokumen Pertama Berdasarkan 11 Data Uji Coba



Gambar 5.5. Nilai Recall Pada 20 Dokumen Pertama Berdasarkan 11 Data Uji Coba



Gambar 5.6. Nilai Recall Pada 30 Dokumen Pertama Berdasarkan 11 Data Uji Coba

Perbedaan nilai *precision* dan *recall* yang tidak berbeda jauh antara pendekatan *fuzzy* yang menggunakan parameter *norm* dengan yang tidak menggunakan parameter *norm*. Kemudahan dalam proses implementasi dapat dicapai dengan menghapus parameter *norm* dalam metode pembobotan.

5.5. Skenario Pengujian 2: Perbandingan Bentuk Rule Fuzzy Terhadap Hasil Akurasi Pencarian Artikel

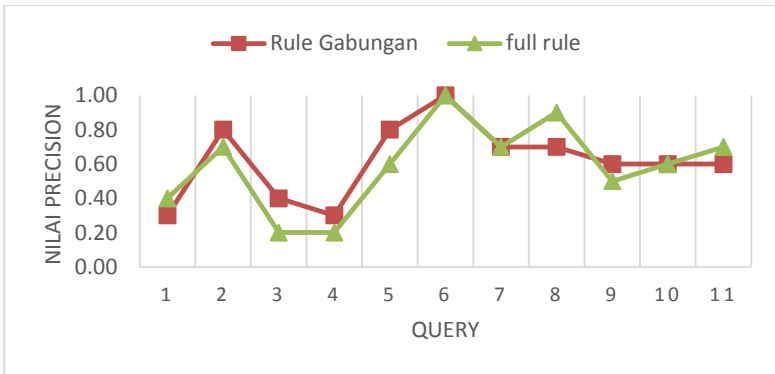
Pada subbab ini akan dijelaskan perbandingan hasil akurasi pencarian artikel antara pendekatan *fuzzy* yang menggunakan seluruh kemungkinan *rule* dengan pendekatan *fuzzy* yang menggunakan beberapa *rule* yang sudah digabungkan. Kedua pendekatan *fuzzy* yang dilakukan menggunakan *membership function* yang sama, perbedaannya ada pada bentuk *rule* yang digunakan. Pada pendekatan pertama akan digunakan seluruh kemungkinan *rule* yang ada seperti yang sudah dijelaskan pada Tabel 3.13, sedangkan pada pendekatan kedua akan digunakan *rule-rule* yang ditunjukkan pada Tabel 3.14 dan Tabel 3.15. Hasil perbandingan nilai rata-rata *precision* dan *recall* dari *fuzzy* yang

menggunakan seluruh kemungkinan *rule* dengan *fuzzy* yang menggunakan beberapa *rule* hasil penggabungan ditunjukkan pada Tabel 5.3.

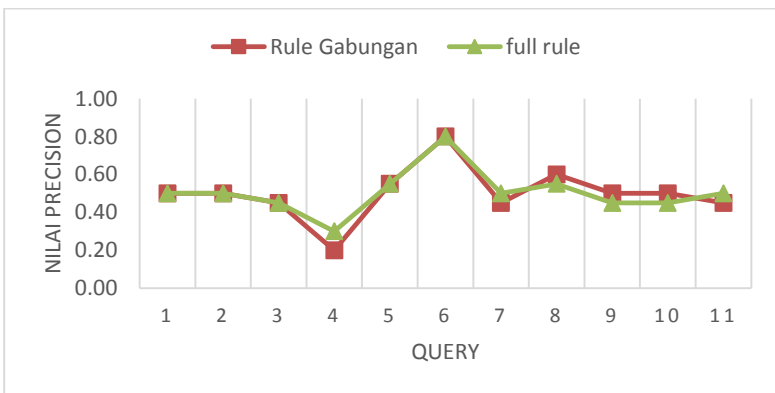
Tabel 5.3. Perbandingan Rata-Rata Precision dan Recall Antara Fuzzy Full Rule dengan Fuzzy Rule Gabungan

Metode Fuzzy	10 Dokumen Pertama		20 Dokumen Pertama		30 Dokumen Pertama		Time (s)
	Precision	Recall	Precision	Recall	Precision	Recall	
Rule Gabungan	0.6182	0.3926	0.5000	0.6361	0.3848	0.7262	1.8146
Full Rule	0.5909	0.3777	0.5045	0.6424	0.3788	0.7198	1.8737

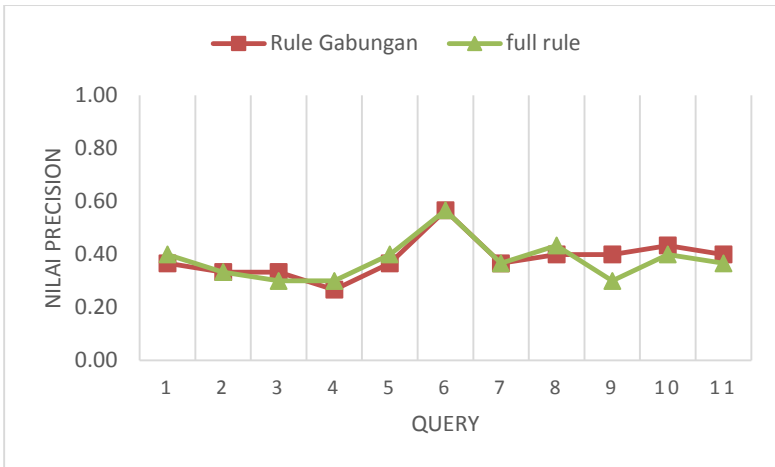
Dari tabel diatas dapat dilihat perbedaan *precision* dan *recall* antara pendekatan *fuzzy* yang seluruh *rule* dan *fuzzy* yang menggunakan *rule* yang sudah digabungkan cukup jauh jika dilihat pada 10 dokumen pertama yang berhasil diambil. Dari 10 dokumen pertama yang berhasil diambil terdapat perbedaan nilai sebesar 0.0273 pada *precision* dan 0.0149 pada *recall*. Perbedaan mulai berkurang jika dilihat pada 20 dokumen pertama yang berhasil diambil, terdapat perbedaan nilai sebesar 0.0045 pada *precision* dan 0.0063 pada *recall*. Perbedaan yang kecil juga terlihat pada 30 dokumen pertama yang berhasil diambil. Pada 30 dokumen pertama yang berhasil diambil, terdapat perbedaan nilai sebesar 0.0060 pada *precision* dan 0.0064 pada *recall*. Grafik yang menunjukkan hasil perhitungan *precision* dan *recall* pada tiap *query* ditunjukkan pada Gambar 5.7 hingga Gambar 5.12. Tiap grafik menggambarkan hubungan antara nilai *precision* atau *recall* (sumbu Y) dengan setiap *query* uji coba (sumbu X).



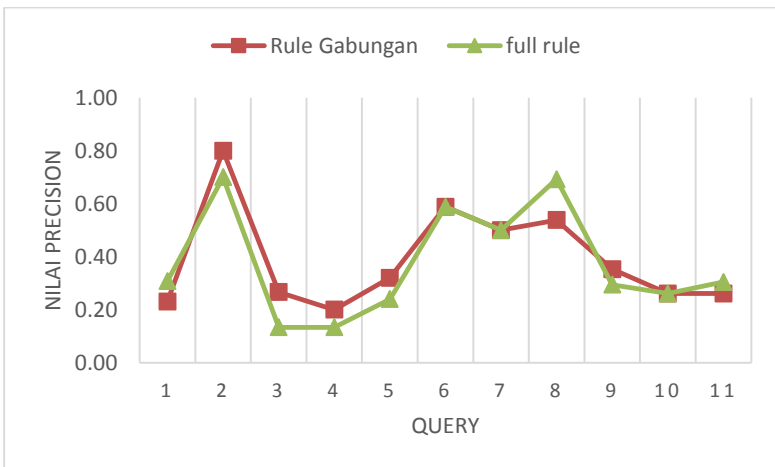
Gambar 5.7. Nilai Precision Pada 10 Dokumen Pertama Berdasarkan 11 Data Uji Coba



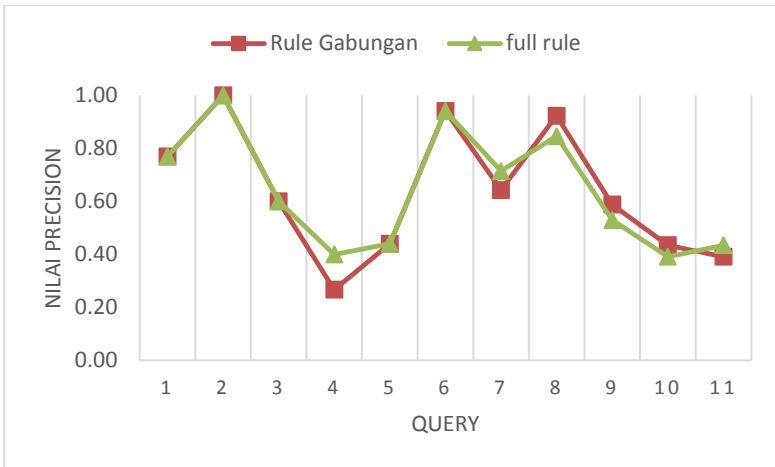
Gambar 5.8. Nilai Precision Pada 20 Dokumen Pertama Berdasarkan 11 Data Uji Coba



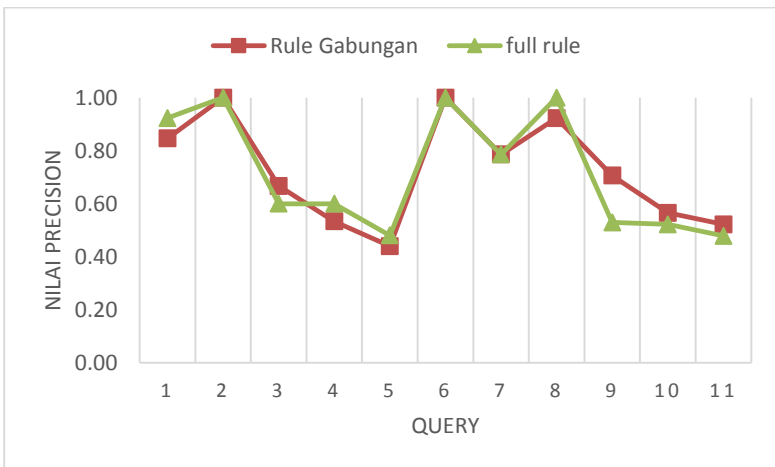
Gambar 5.9. Nilai Precision Pada 30 Dokumen Pertama Berdasarkan 11 Data Uji Coba



Gambar 5.10. Nilai Recall Pada 10 Dokumen Pertama Berdasarkan 11 Data Uji Coba



Gambar 5.11. Nilai Recall Pada 20 Dokumen Pertama Berdasarkan 11 Data Uji Coba



Gambar 5.12. Nilai Recall Pada 30 Dokumen Pertama Berdasarkan 11 Data Uji Coba

Perbedaan nilai *precision* dan *recall* yang tidak berbeda jauh antara pendekatan *fuzzy* yang menggunakan seluruh *rule* dengan

yang menggunakan *rule* gabungan. Kemudahan dalam proses implementasi dapat dicapai dengan menggunakan 27 *rule* hasil penggabungan dari 125 *rule* yang ada.

5.6.Skenario Pengujian 3: Perbandingan Penggunaan FLC Query Terhadap Hasil Akurasi Pencarian Artikel

Pada subbab ini akan dijelaskan perbandingan hasil akurasi pencarian artikel antara pendekatan *fuzzy* yang menggunakan FLC *query* dalam prosesnya dengan pendekatan *fuzzy* yang tidak menggunakan FLC *query* dalam prosesnya. Penghapusan FLC *query* dari proses pembobotan mengakibatkan hanya digunakan FLC *document* dan mengubah struktur *fuzzy* menjadi 1 tingkat. Hasil perbandingan nilai rata-rata *precision* dan *recall* dari kedua pendekatan ditunjukkan pada Tabel 5.4

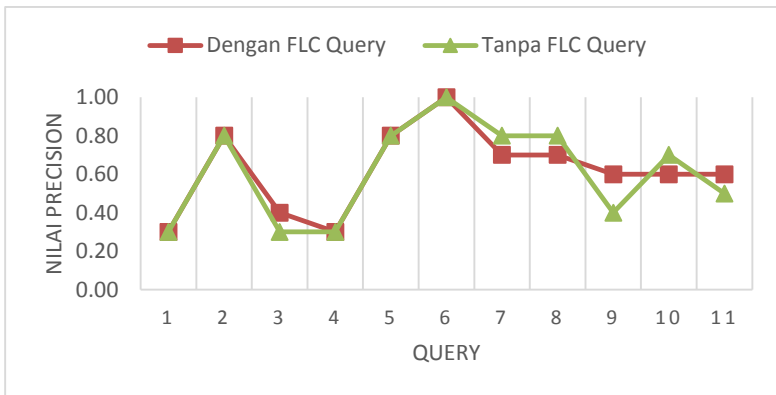
Tabel 5.4. Rata-Rata Precision dan Recall dari Pendekatan yang Menggunakan FLC Query dan Tanpa FLC Query

Metode Fuzzy	10 Dokumen Pertama		20 Dokumen Pertama		30 Dokumen Pertama		Time (s)
	Precision	Recall	Precision	Recall	Precision	Recall	
Dengan FLC Query	0.6182	0.3926	0.5000	0.6361	0.3848	0.7262	1.8146
Tanpa FLC Query	0.6091	0.3809	0.4727	0.6247	0.3909	0.7517	1.5930

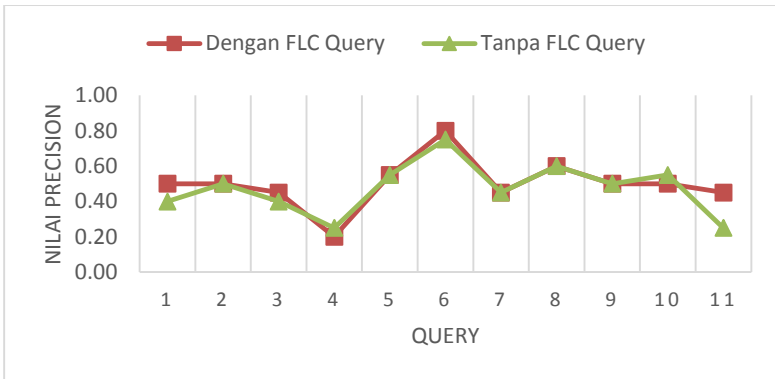
Dari tabel diatas dapat dilihat perbedaan *precision* dan *recall* antara metode yang menggunakan FLC *query* dan yang tidak menggunakan FLC *query* tidak berbeda jauh jika dilihat pada 10 dokumen pertama yang berhasil diambil.

Dari 10 dokumen pertama yang berhasil diambil terdapat perbedaan nilai sebesar 0.0091 pada *precision* dan 0.0117 pada

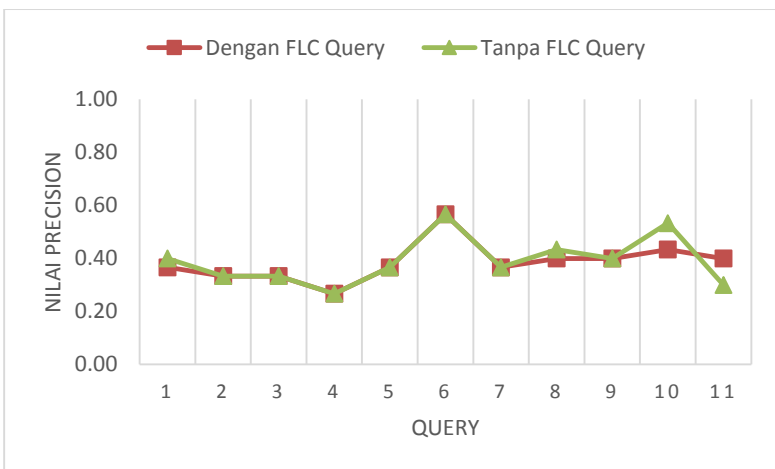
recall. Perbedaan yang tidak besar juga terlihat pada 20 dokumen pertama yang berhasil diambil, terdapat perbedaan nilai sebesar 0.0273 pada *precision* dan 0.0114 pada *recall*. Perbedaan yang tidak besar juga terlihat pada 30 dokumen pertama yang berhasil diambil. Pada 30 dokumen pertama yang berhasil diambil, terdapat perbedaan nilai sebesar 0.0061 pada *precision* dan 0.0255 pada *recall*. Grafik yang menunjukkan hasil perhitungan *precision* dan *recall* pada tiap *query* ditunjukkan pada Gambar 5.13 hingga Gambar 5.18. Tiap grafik menggambarkan hubungan antara nilai *precision* atau *recall* (sumbu Y) dengan setiap *query* uji coba (sumbu X).



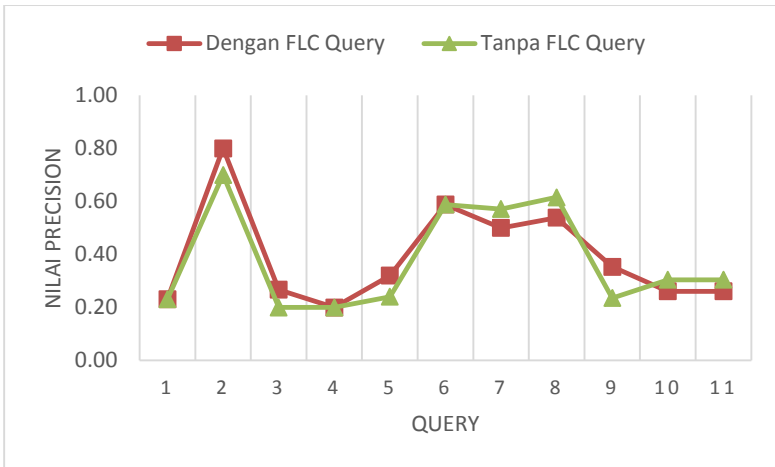
Gambar 5.13. Nilai Precision Pada 10 Dokumen Pertama Berdasarkan 11 Data Uji Coba



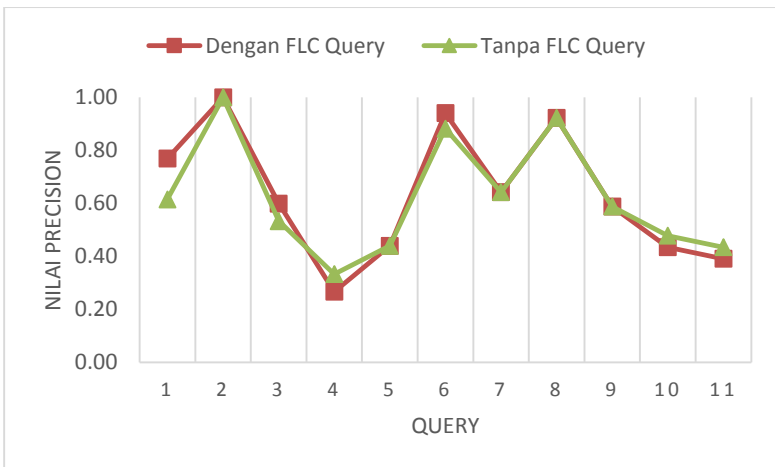
Gambar 5.14. Nilai Precision Pada 20 Dokumen Pertama Berdasarkan 11 Data Uji Coba



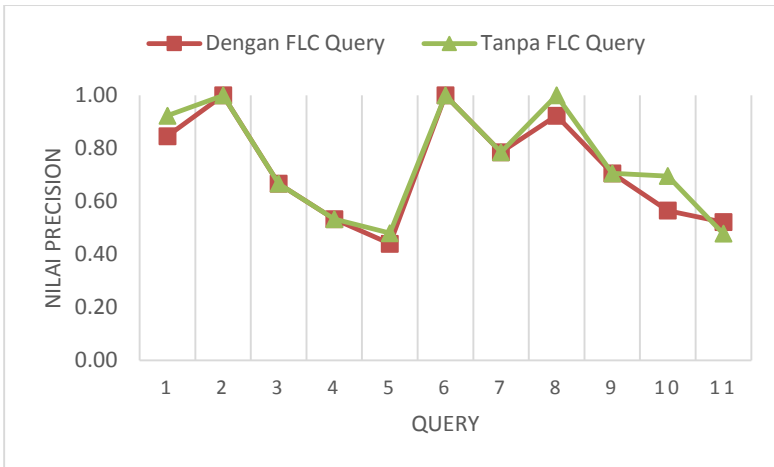
Gambar 5.15. Nilai Precision Pada 30 Dokumen Pertama Berdasarkan 11 Data Uji Coba



Gambar 5.16. Nilai Recall Pada 10 Dokumen Pertama Berdasarkan 11 Data Uji Coba



Gambar 5.17. Nilai Recall Pada 20 Dokumen Pertama Berdasarkan 11 Data Uji Coba



Gambar 5.18. Nilai Recall Pada 30 Dokumen Pertama Berdasarkan 11 Data Uji Coba

Perbedaan nilai *precision* dan *recall* yang tidak berbeda jauh antara pendekatan *fuzzy* yang menggunakan FLC *query* dengan yang tidak menggunakan FLC *query*. Kemudahan dalam proses implementasi dapat dicapai dengan menghapus penggunaan FLC *query* dalam metode pembobotan.

5.7. Skenario Pengujian 4: Perbandingan Pendekatan Fuzzy dan Non Fuzzy Terhadap Hasil Akurasi Pencarian Artikel

Pada subbab ini akan dijelaskan perbandingan hasil akurasi pencarian artikel antara metode yang menggunakan pendekatan *fuzzy* dengan metode *non fuzzy* (TF – IDF) yang dijelaskan pada subbab 2.3. Hasil perbandingan nilai rata-rata *precision* dan *recall* dari kedua metode ditunjukkan pada Tabel 5.5

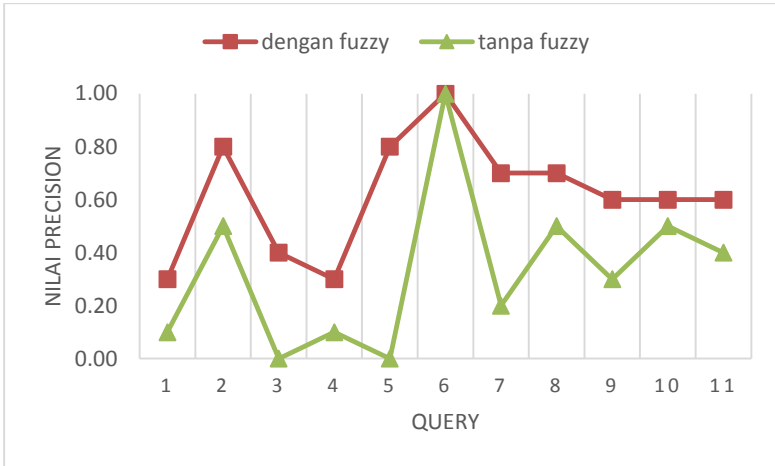
Tabel 5.5. Perbandingan Rata-Rata Precision dan Recall Antara Metode dengan Pendekatan Fuzzy dan Non Fuzzy

Metode Fuzzy	10 Dokumen Pertama		20 Dokumen Pertama		30 Dokumen Pertama		Time (s)
	Precision	Recall	Precision	Recall	Precision	Recall	
Fuzzy	0.6182	0.3926	0.5000	0.6361	0.3848	0.7262	1.8146
Non Fuzzy	0.3273	0.2116	0.3091	0.4008	0.2848	0.5432	1.8873

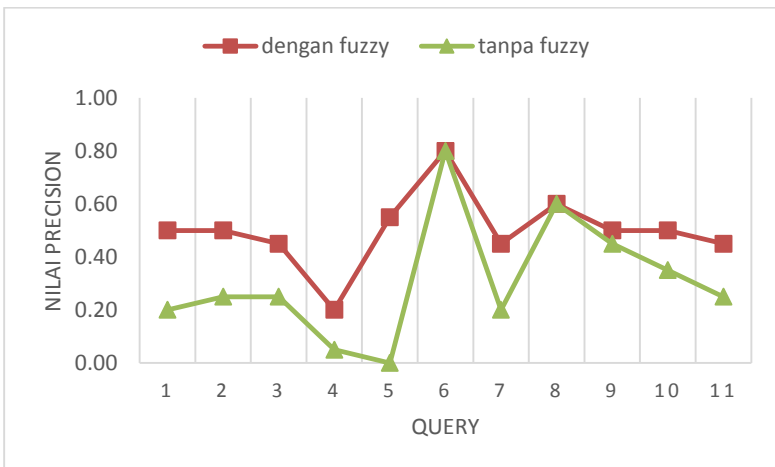
Dari tabel diatas dapat dilihat perbedaan *precision* dan *recall* antara metode yang menggunakan pendekatan *fuzzy* dan metode yang tanpa *fuzzy* sangat jauh jika dilihat pada 10 dokumen pertama yang berhasil diambil.

Dari 10 dokumen pertama yang berhasil diambil terdapat perbedaan nilai sebesar 0.2909 pada *precision* dan 0.1810 pada *recall*. Perbedaan yang sangat besar juga terlihat pada 20 dokumen pertama yang berhasil diambil, terdapat perbedaan nilai sebesar 0.1909 pada *precision* dan 0.2353 pada *recall*. Perbedaan yang cukup besar juga terlihat pada 30 dokumen pertama yang berhasil diambil. Pada 30 dokumen pertama yang berhasil diambil, terdapat perbedaan nilai sebesar 0.1000 pada *precision* dan 0.1830 pada *recall*. Grafik yang menunjukkan hasil perhitungan *precision* dan *recall* pada tiap *query* ditunjukkan pada Gambar 5.19 hingga Gambar 5.24. Tiap grafik menggambarkan hubungan antara nilai

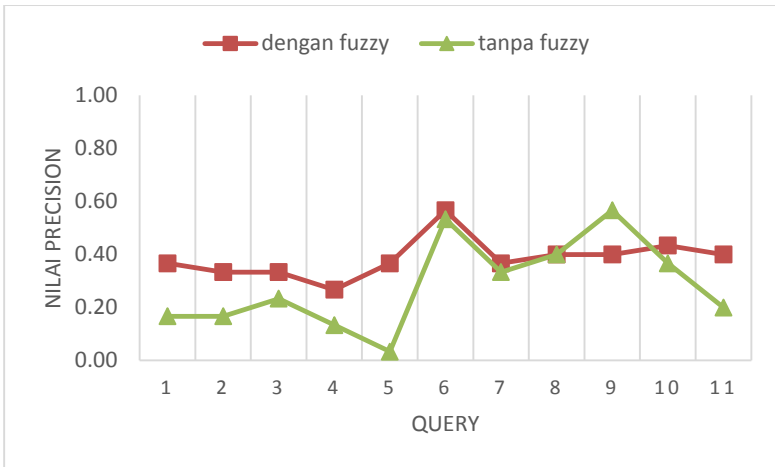
precision atau *recall* (sumbu Y) dengan setiap *query* uji coba (sumbu X).



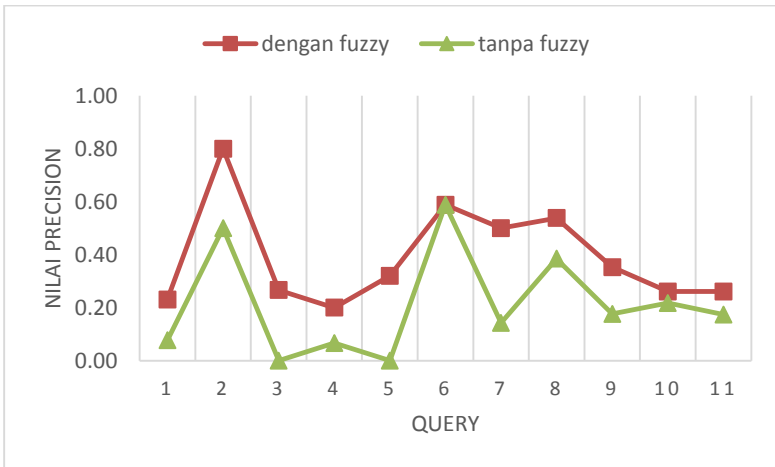
Gambar 5.19. Nilai Precision Pada 10 Dokumen Pertama Berdasarkan 11 Data Uji Coba



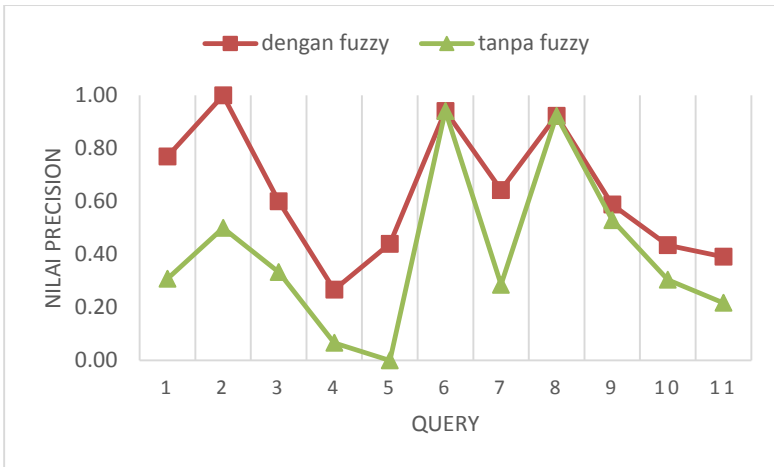
Gambar 5.20. Nilai Precision Pada 20 Dokumen Pertama Berdasarkan 11 Data Uji Coba



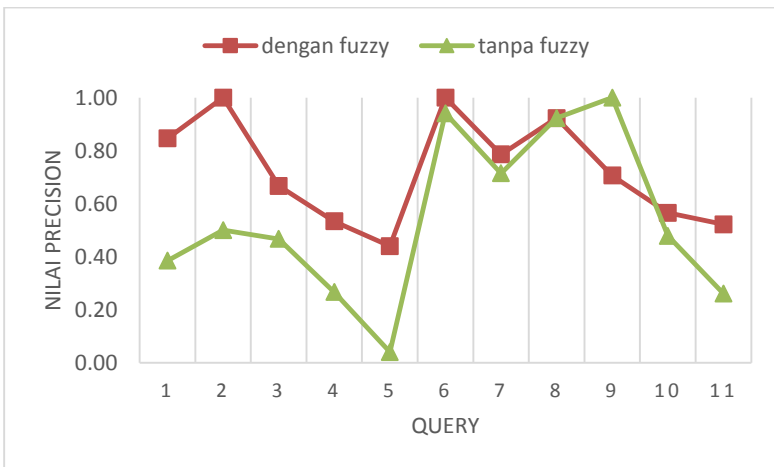
Gambar 5.21. Nilai Precision Pada 30 Dokumen Pertama Berdasarkan 11 Data Uji Coba



Gambar 5.22. Nilai Recall Pada 10 Dokumen Pertama Berdasarkan 11 Data Uji Coba



Gambar 5.23. Nilai Recall Pada 20 Dokumen Pertama Berdasarkan 11 Data Uji Coba



Gambar 5.24. Nilai Recall Pada 30 Dokumen Pertama Berdasarkan 11 Data Uji Coba

Pada skenario 4 ini dapat dilihat bahwa pendekatan *fuzzy* menghasilkan nilai *precision* dan *recall* yang jauh lebih baik jika

dibandingkan dengan yang menggunakan pendekatan TF – IDF standar. Contoh hasil pencarian dengan kedua pendekatan ditunjukkan pada Tabel 5.6. Judul artikel yang diberi blok berwarna hijau merupakan artikel yang relevan dengan *query*.

Tabel 5.6. Hasil Pencarian dengan Pendekatan Fuzzy dan Non Fuzzy pada Query Wisata Bahari Indonesia

No	Fuzzy	Non Fuzzy
1	Wisata Karimunjawa, Satu Destinasi Bahari Menawan di Indonesia	Wisata Bahari Lamongan
2	Wisata Bahari Lamongan	Tempat Menarik Di Indonesia
3	Jelajah Pariwisata Indonesia	Wisata Karimunjawa, Satu Destinasi Bahari Menawan di Indonesia
4	Kepulauan Derawan, Pesona Wisata Bahari Kaltim	Wisata Indonesia Yang Menarik
5	Wisata Gili Trawangan, Pesona Bahari Lombok yang Menawan	Objek Wisata di Indonesia
6	Wisata Bahari Lamongan, Destinasi Rekreasi Favorit di Jawa Timur	destinasi wisata Indonesia
7	Wisata Tanjung Lesung, Destinasi Bahari Banten yang Memikat	wisata indonesia yang belum terjamah
8	wisata indonesia yang belum terjamah	Resep Masakan Indonesia
9	Wisata Banda Aceh yang Menarik dan Patut Dikunjungi	Jelajah Pariwisata Indonesia
10	Pantai Selong Belanak Lombok, Wisata Bahari Penuh Kedamaian	Pantai Terbaik Di Indonesia

No	Fuzzy	Non Fuzzy
11	Kepulauan Alor, Wisata Bahari NTT	Info Pariwisata Indonesia
12	Inilah 8 Tempat Wisata di Manado Paling Menarik	Surga Tersembunyi Di Indonesia
13	Wisata Indonesia Yang Menarik	Pantai Terindah Indonesia
14	Wisata Indonesia yang Wajib Dikunjungi	Bali, Surga di Indonesia
15	WISATA INDONESIA, WISATA JOGJAKARTA	WISATA INDONESIA, WISATA JOGJAKARTA
16	Inilah 12 Tempat Wisata di Jogja Paling Terkenal	Kepulauan Alor, Wisata Bahari NTT
17	Wisata Bunaken, Pesona Taman Laut Manado Terpopuler	Wisata Indonesia yang Wajib Dikunjungi
18	Inilah 10 Tempat Wisata di Lombok Paling Menawan	Wisata Indonesia Yang Jarang Dikunjungi
19	Inilah Pesona 8 Tempat Wisata di Jepara Paling Menarik	WISATA INDONESIA , KUTA LOMBOK
20	Inilah 7 Tempat Wisata di Makassar Paling Populer	Wisata Indonesia : Raja Ampat
21	Wisata kepulauan Seribu	Kepulauan Derawan, Pesona Wisata Bahari Kaltim
22	Objek Wisata di Indonesia	KERAJINAN TANGAN KHAS INDONESIA
23	destinasi wisata indonesia	Wisata Bahari Lamongan, Destinasi Rekreasi Favorit di Jawa Timur
24	Wisata Indonesia Yang Jarang Dikunjungi	Wisata Gili Trawangan, Pesona Bahari Lombok yang Menawan
25	WISATA INDONESIA , KUTA LOMBOK	Wisata Tanjung Lesung, Destinasi Bahari Banten yang Memikat

No	Fuzzy	Non Fuzzy
26	Menikmati Alam Indonesia di 4 Tempat Wisata Temanggung	Pantai Baron Gunungkidul, Kuliner Bahari Jogja Paling Menarik
27	Mengunjungi Tempat Wisata di Sukabumi Jawa Barat	10 Tempat Wisata Alam di Indonesia yang Wajib Dikunjungi
28	Wisata Kebumen	Menikmati Alam Indonesia di 4 Tempat Wisata Temanggung
29	Diving Raja Ampat, Surganya Wisata Laut Indonesia	7 Resort Cantik Tempat Liburan Wisata di Indonesia
30	6 Spot Wisata Pelabuhan Ratu di Sukabumi Terpopuler	Pantai Selong Belanak Lombok, Wisata Bahari Penuh Kedamaian

Pada tabel diatas ditunjukkan terdapat banyak artikel tidak relevan yang berada pada urutan atas jika menggunakan pembobotan TF-IDF standar (tanpa *fuzzy*). Hal ini dikarenakan pada pembobotan tersebut, banyak artikel yang memiliki nilai *term frequency* yang tinggi terhadap *term-term* yang umum seperti 'wisata' dan 'indonesia'. Nilai *inverse document frequency* yang tinggi tidak terlalu berpengaruh jika nilai *term frequency* terlalu rendah. Sebuah artikel yang memiliki nilai *term frequency* dan *inverse document frequency* yang berbeda jauh cenderung merupakan artikel yang tidak relevan.

Pada pembobotan dengan pendekatan *fuzzy*, nilai *term frequency* dan *inverse document frequency* akan dipetakan kedalam nilai *fuzzy* sesuai dengan *membership function* dan akan dilakukan operasi sesuai *rule-rule* yang sudah ditentukan. Dengan adanya operasi menggunakan *rule-rule fuzzy* ini, dapat dihasilkan pembobotan yang lebih efektif.

Pada Tabel 5.7 akan ditunjukkan perbandingan hasil pencarian yang dihasilkan oleh metode *fuzzy* dan *non fuzzy* jika menggunakan *query* 'Wisata Lombok'. Judul artikel yang diberi blok berwarna hijau merupakan artikel yang relevan dengan *query*.

Tabel 5.7. Hasil Pencarian dengan Pendekatan Fuzzy dan Non Fuzzy pada Query Wisata Lombok

No	Fuzzy	Non Fuzzy
1	Tempat Wisata Di Lombok	Tempat Wisata Di Lombok
2	Wisata Kuliner Lombok	Wisata Kuliner Lombok
3	Wisata Gili Trawangan, Pesona Bahari Lombok yang Menawan	Gili Air Lombok
4	Inilah 10 Tempat Wisata di Lombok Paling Menawan	Inilah 10 Tempat Wisata di Lombok Paling Menawan
5	WISATA INDONESIA , KUTA LOMBOK	WISATA INDONESIA , KUTA LOMBOK
6	10 Tempat Wisata di Lombok yang Wajib Dikunjungi	Pesona Gili Trawangan Lombok
7	Wisata Gili Trawangan, Pulau di Lombok yang Seru dan Cantik	Keindahan Gili Meno Lombok
8	Wisata Gunung Rinjani Lombok, Lanskap Eksotis dari Ketinggian	10 Tempat Wisata di Lombok yang Wajib Dikunjungi
9	10 Tempat Wisata Pantai di Lombok yang Wajib Dikunjungi	Menikmati Sunset di Pantai Senggigi Lombok
10	Pantai Selong Belanak Lombok, Wisata Bahari Penuh Kedamaian	Pesona Menakjubkan di Pantai Senggigi Lombok
11	Pesona Menakjubkan di Pantai Senggigi Lombok	10 Tempat Wisata Pantai di Lombok yang Wajib Dikunjungi
12	Pesona Gili Trawangan Lombok	Lanskap Indah Pantai Kuta Lombok dari Bukit Mandalika
13	Keindahan Gili Meno Lombok	Wisata Gunung Rinjani Lombok, Lanskap Eksotis dari Ketinggian

No	Fuzzy	Non Fuzzy
14	Gili Air Lombok	Wisata Gili Trawangan, Pesona Bahari Lombok yang Menawan
15	Menikmati Sunset di Pantai Senggigi Lombok	Wisata Gili Trawangan, Pulau di Lombok yang Seru dan Cantik
16	Lanskap Indah Pantai Kuta Lombok dari Bukit Mandalika	Pantai Selong Belanak Lombok, Wisata Bahari Penuh Kedamaian
17	Inilah 8 Tempat Wisata di Manado Paling Menarik	Tempat Wisata Purwokerto
18	Pesona Diving di Tempat Wisata Pulau Weh Sabang	Tempat wisata di banten
19	4 Tempat Wisata Pantai di Jakarta Terpopuler	Wisata Karimunjawa
20	10 Tempat Wisata Kuliner di Semarang yang Wajib Dikunjungi	Wisata Jepara
21	Wisata Indonesia Yang Jarang Dikunjungi	Tempat Wisata DI Bogor
22	Wisata Gunung Rinjani	Tempat Wisata Di Penang
23	Wisata Terbaik Jawa Timur	Wisata Klaten
24	Objek Wisata di Indonesia	Wisata Purworejo
25	10 Tempat Wisata Alam di Indonesia yang Wajib Dikunjungi	Tempat Wisata Di Tuban
26	Rekomendasi Tempat Wisata Kuliner Semarang Terfavorit	Wisata Cianjur
27	Tempat wisata di banten	Wisata Ubud
28	Wisata Blitar	Tempat wisata di malaysia
29	Tempat wisata di malaysia	Tempat Wisata di Sumedang
30	Tempat Wisata Di Banyuwangi	Tempat Wisata Probolinggo

Pada tabel diatas dapat dilihat pembobotan yang menggunakan pendekatan *fuzzy* dan yang tidak menggunakan pendekatan *fuzzy*

memberikan hasil yang relatif sama jika dilakukan pencarian dengan *query* ‘Wisata Lombok’. Sehingga dapat disimpulkan bahwa hasil perbandingan antara pembobotan yang menggunakan pendekatan *fuzzy* dengan yang tidak menggunakan *fuzzy* memberikan hasil yang sama pada *query* yang mengandung sedikit kata tidak penting.

5.8. Analisa Hasil Uji Coba

Pada subbab ini akan dijelaskan analisa dari hasil uji coba yang sudah dilakukan. Analisa ini akan dilakukan pada tiap skenario uji coba yang sudah dilakukan.

Skenario 1 melakukan analisa perbandingan hasil pembobotan berdasarkan parameter yang dipakai dalam pendekatan *fuzzy*. Pendekatan *fuzzy* yang menggunakan 3 parameter (*term frequency*, *inverse document frequency*, dan *norm*) memberikan hasil pembobotan yang tidak berbeda jauh dengan pendekatan *fuzzy* yang menggunakan 2 parameter (*term frequency* dan *inverse document frequency*) seperti yang ditunjukkan pada Tabel 5.2. Meski demikian, pendekatan *fuzzy* yang menggunakan 3 parameter tetap menghasilkan nilai *precision* dan *recall* yang sedikit lebih baik jika dibandingkan dengan pendekatan *fuzzy* yang menggunakan 2 parameter. Skenario 1 menunjukkan bahwa parameter *term frequency* dan *inverse document frequency* memiliki peran yang lebih penting jika dibandingkan dengan parameter *norm*, sehingga penghapusan parameter *norm* tidak memberikan perubahan yang besar terhadap hasil pembobotan.

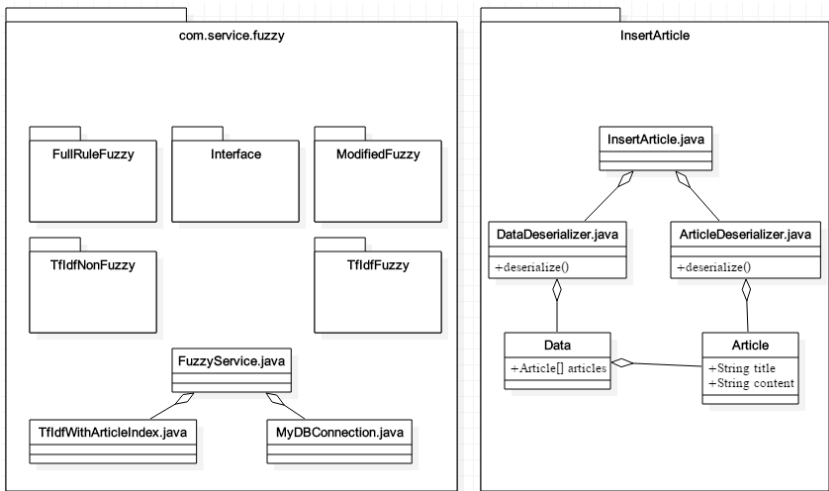
Skenario 2 melakukan analisa perbandingan hasil pembobotan berdasarkan *rule fuzzy* yang digunakan. Pendekatan *fuzzy* yang pertama menggunakan seluruh kemungkinan *rule fuzzy* yang ada, kemudian pendekatan *fuzzy* selanjutnya menggunakan *rule-rule fuzzy* yang merupakan gabungan dari seluruh kemungkinan *rule* yang ada. Skenario 2 menunjukkan bahwa hasil pendekatan dengan 2 jenis *rule* tersebut memberikan hasil yang tidak berbeda jauh seperti yang ditunjukkan pada Tabel 5.3, sehingga dapat

disimpulkan *rule* yang merupakan gabungan dari seluruh kemungkinan *rule* memberikan hasil yang relatif sama.

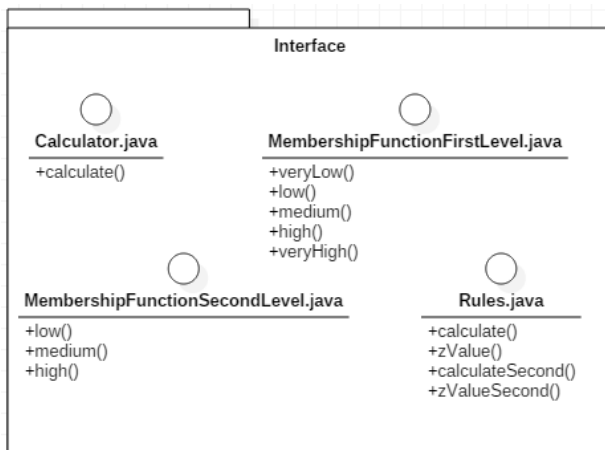
Skenario 3 melakukan analisa perbandingan hasil pembobotan berdasarkan penggunaan FLC *query*. Pendekatan *fuzzy* yang pertama menggunakan FLC *query* didalam prosesnya, sedangkan pendekatan *fuzzy* yang kedua tidak menggunakan FLC *query* didalam prosesnya. Skenario 3 menunjukkan tidak adanya perbedaan yang signifikan pada nilai *precision* dan *recall* jika menggunakan FLC *query* dalam pendekatan *fuzzy* seperti yang ditunjukkan pada Tabel 5.4.

Skenario 4 melakukan analisa perbandingan hasil pembobotan yang menggunakan pendekatan *fuzzy* dan pembobotan TF – IDF standar. Skenario 4 menunjukkan terdapat peningkatan *precision* dan *recall* yang sangat signifikan pada pembobotan yang menggunakan pendekatan *fuzzy* jika dibandingkan dengan pendekatan *non fuzzy* seperti yang ditunjukkan pada Tabel 5.5. Pembobotan TF – IDF dilakukan dengan mengalikan nilai *term frequency* dengan *inverse document frequency* secara langsung. Sedangkan dalam pembobotan dengan pendekatan *fuzzy*, nilai *term frequency* dan *inverse document frequency* akan diproses menggunakan logika *fuzzy*. Dalam pembobotan dengan pendekatan *fuzzy* juga ditambahkan parameter *norm* untuk meningkatkan hasil pembobotan. Dengan demikian dapat disimpulkan pembobotan dengan menggunakan pendekatan *fuzzy* jauh lebih baik dalam hal *precision* dan *recall* jika dibandingkan dengan pembobotan *non fuzzy* (TF – IDF).

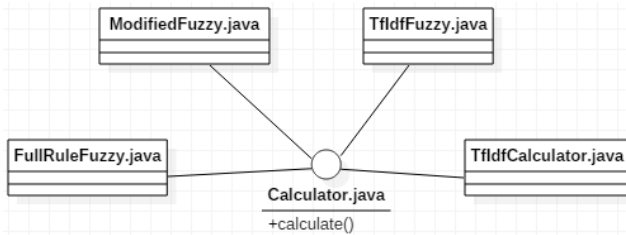
LAMPIRAN 1: Modul Sistem



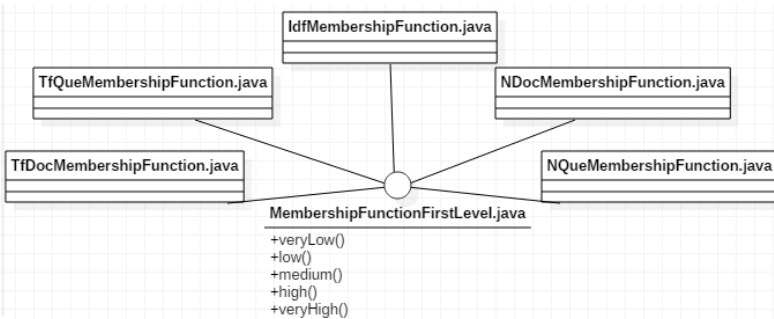
Gambar 8.1. Keseluruhan Modul Sistem Pencarian Artikel Pariwisata



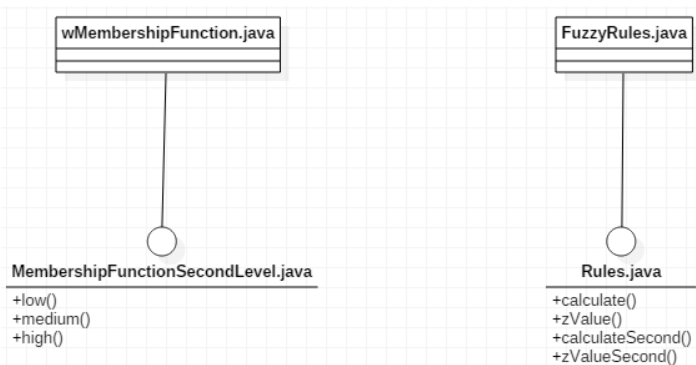
Gambar 8.2. Isi Package Interface



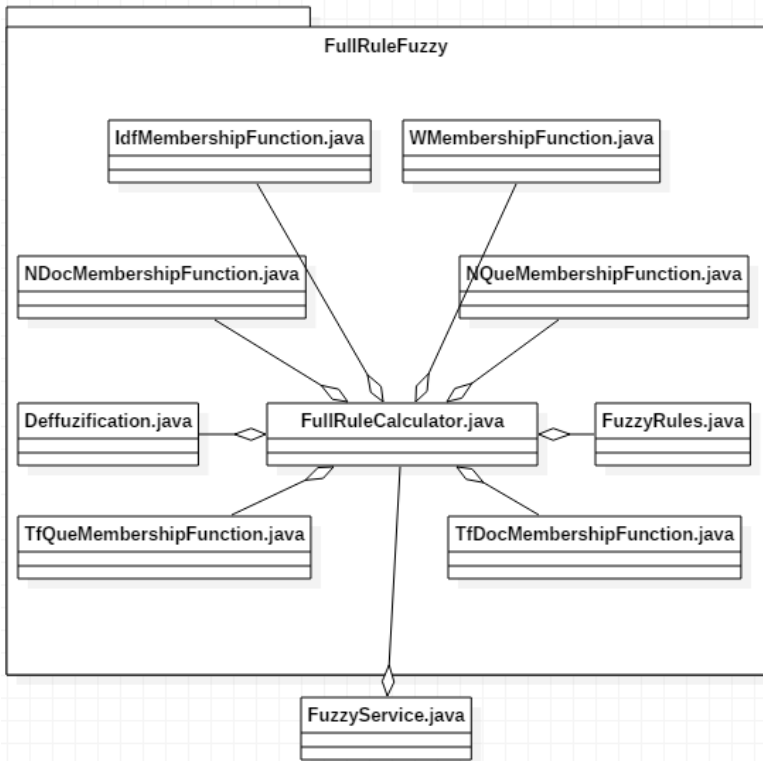
Gambar 8.3. Kelas yang Mengimplementasikan Interface Calculator



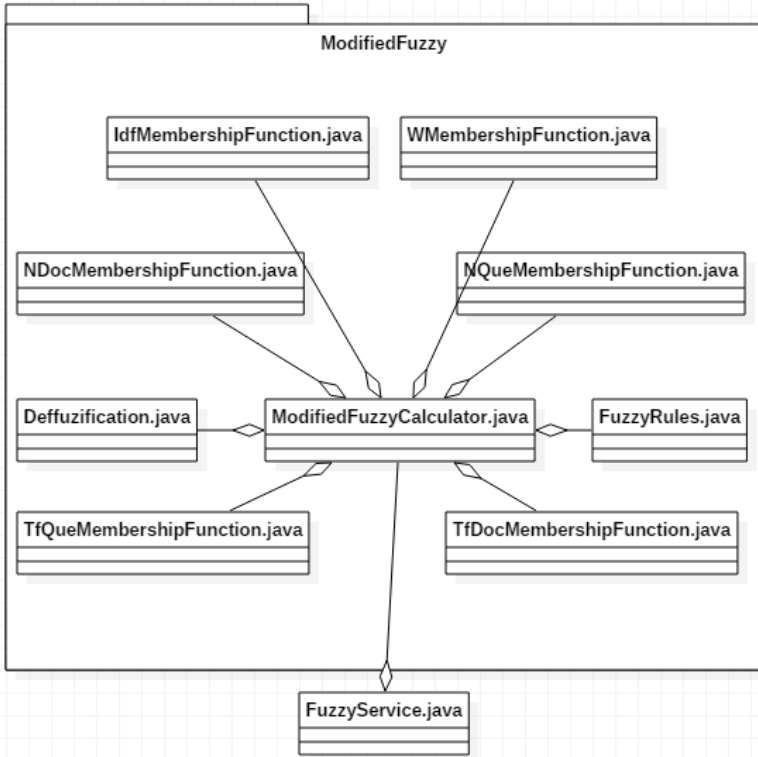
Gambar 8.4. Kelas yang Mengimplementasikan Interface MembershipFunctionFirstLevel



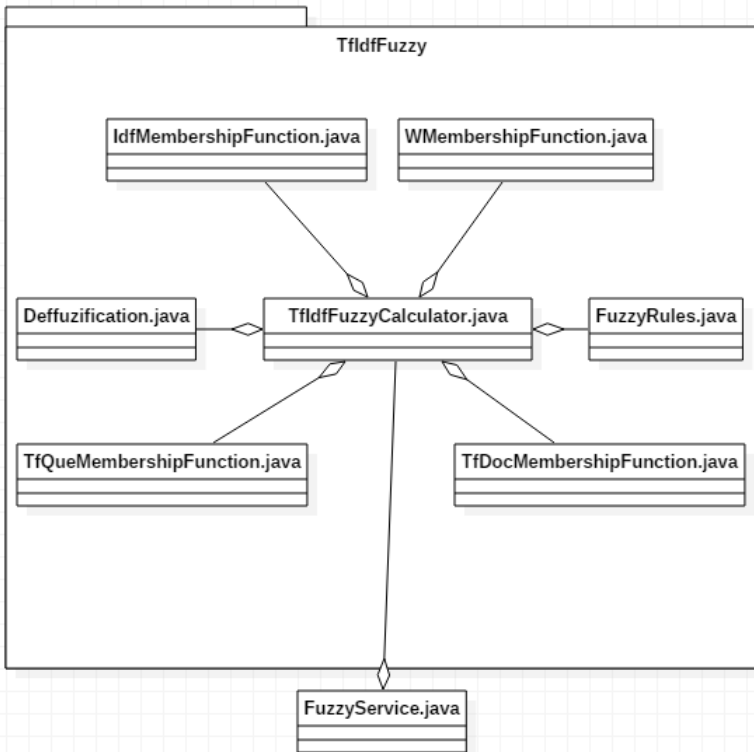
Gambar 8.5. Kelas yang Mengimplementasikan Interface MembershipFunctionSecondLevel dan Rules



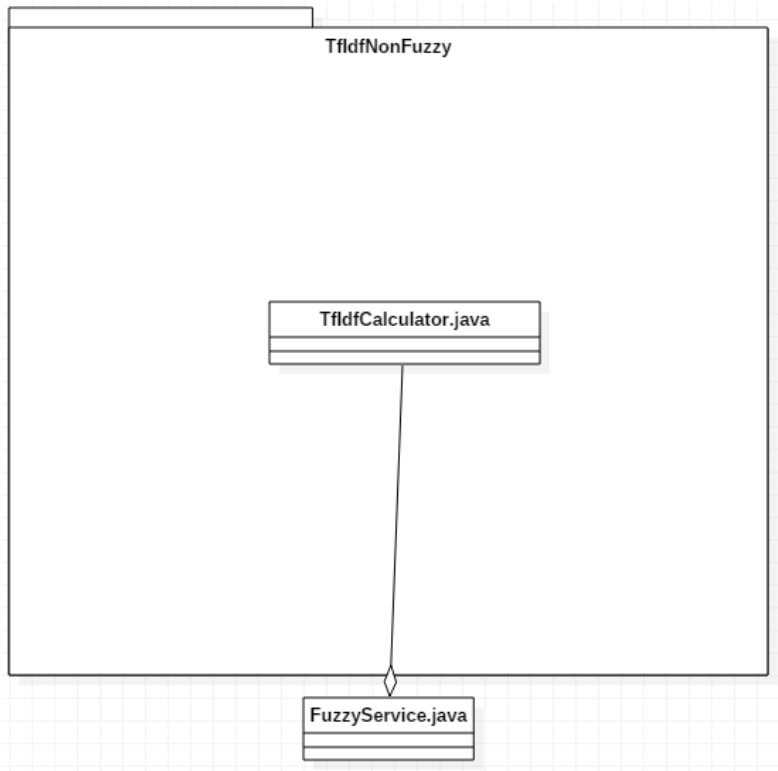
Gambar 8.6. Isi Package FullRuleFuzzy



Gambar 8.7. Isi Package ModifiedFuzzy



Gambar 8.8. Isi Package TfldfFuzzy



Gambar 8.9. Isi Package TfidfNonFuzzy

LAMPIRAN 2: Daftar Query Uji Coba dan Artikel Relevan

Tabel 9.1. Daftar Query Uji Coba dan Artikel Relevan

No	Query	Artikel Relevan
1	Pantai Jawa Timur	10 Pantai Eksotis di Jawa Timur yang Wajib Dikunjungi
		10 Tempat Wisata Alam di Jawa Timur yang Wajib Dikunjungi
		Wisata Bahari Lamongan, Destinasi Rekreasi Favorit di Jawa Timur
		Pantai Popoh Tulungagung, Keindahan dan Kuliner Lautnya yang Lezat
		Pantai Klayar Pacitan, Eksotisme Alam yang Begitu Kental
		Inilah Pantai Pulau Merah di Banyuwangi Yang Memikat Hati
		Pantai Plengkung Banyuwangi, Destinasi Favorit Para Surfer Dunia
		Pantai Pasir Putih Situbondo, Salah Satu yang Populer di Jatim
		Pantai Tambakrejo Blitar, Satu Pilihan Destinasi Wisata yang Menarik
		Pantai Teleng Ria Pacitan, Horizon Indah dari Perbukitan
		Pantai Bale Kambang
		Pantai Pelang Trenggalek, Lanskap Pantai yang Indah hingga Air Terjun
		Wisata Pantai Pacitan
		Wisata Kuliner di Bandung
2	Wisata Kuliner Bandung	Tempat Wisata Kuliner di Bandung
		Rekomendasi Tempat Wisata Kuliner Bandung Favorit

No	Query	Artikel Relevan
		Floating Market Lembang Bandung, Kuliner Lezat di Pasar Terapung Unik
		15 Tempat Wisata Kuliner di Bandung yang Wajib Dikunjungi
		Ciwalk Bandung, Destinasi Belanja hingga Kuliner di Kota Kembang
		Wisata di Lembang Bandung
		8 Tempat Wisata di Lembang Bandung Terpopuler
		15 Tempat Wisata di Bandung yang Wajib Dikunjungi
		Inilah 20 Tempat Wisata di Bandung Paling Terkenal Untuk Dikunjungi
3	Kuliner Jawa Barat	Wisata Kuliner Cirebon
		Wisata Kuliner di Bogor
		Wisata Kuliner Depok
		Wisata Kuliner Sukabumi
		Wisata Kuliner Tasikmalaya
		Wisata Kuliner Garut
		Inilah Rekomendasi 8 Tempat Kuliner Bogor Favorit
		Wisata Kuliner di Bandung
		Tempat Wisata Kuliner di Bandung
		Rekomendasi Tempat Wisata Kuliner Bandung Favorit
		Floating Market Lembang Bandung, Kuliner Lezat di Pasar Terapung Unik
		15 Tempat Wisata Kuliner di Bandung yang Wajib Dikunjungi
		Ciwalk Bandung, Destinasi Belanja hingga Kuliner di Kota Kembang
		Wisata Kuliner Bekasi

No	Query	Artikel Relevan
4	Kuliner Jawa Tengah	15 Tempat Wisata Kuliner di Bogor yang Wajib Dikunjungi
		Wisata Kuliner Magelang
		Rekomendasi Tempat Wisata Kuliner Purwokerto
		Kuliner Klaten
		Wisata Kuliner Yogyakarta
		Rekomendasi Tempat Wisata Kuliner Semarang Terfavorit
		10 Tempat Wisata Kuliner di Solo yang Wajib Dikunjungi
		10 Tempat Wisata Kuliner di Semarang yang Wajib Dikunjungi
		10 Tempat Wisata Kuliner di Jogja yang Wajib Dikunjungi
		Pantai Baron Gunungkidul, Kuliner Bahari Jogja Paling Menarik
		Wisata Kuliner Purwokerto
		8 Restoran Recommended Tempat Wisata Kuliner Jogja
		Rekomendasi Wisata Kuliner Jogja Terfavorit
		Ini 7 Tempat Kuliner Tradisional Jogja Menarik
		7 Cafe Recommended Tempat Wisata Kuliner Jogja
		Wisata Kuliner Solo
5	Wisata Jogjakarta	WISATA INDONESIA, WISATA JOGJAKARTA
		10 Tempat Wisata Kuliner di Jogja yang Wajib Dikunjungi
		6 Oleh-Oleh Khas Jogja Paling Favorit

No	Query	Artikel Relevan
		7 Cafe Recommended Tempat Wisata Kuliner Jogja
		8 Restoran Recommended Tempat Wisata Kuliner Jogja
		Destinasi Wisata Istimewa Di Jogja
		Goa Pindul Jogja, Satu Permata Wisata Goa yang Menawan
		Goa Selarong, Pesona Religi Wisata Goa di Jogja
		Ini 6 Spot Menarik Wisata Kaliurang di Kaki Merapi Jogja
		Ini 7 Tempat Kuliner Tradisional Jogja Menarik
		Inilah 12 Tempat Wisata di Jogja Paling Terkenal
		Inilah 12 Tempat Wisata Pantai di Jogja Paling Menarik
		Inilah 13 Candi di Jogja Paling Menawan
		Inilah 13 Spot Wisata Gunungkidul Jogja Paling Populer
		Inilah 5 Spot Wisata Kotagede Yogyakarta Paling Menarik
		Inilah Pesona Air Terjun Sri Gethuk di Gunungkidul Jogja
		Objek Wisata Gunung Kidul
		Pantai Baron Gunungkidul, Kuliner Bahari Jogja Paling Menarik
		Pantai Krakal Gunungkidul, Surfing dan Lanskap Alam Indah di Jogja
		Pantai Pok Tunggal Gunungkidul, Sebuah Destinasi di Jogja yang Disukai

No	Query	Artikel Relevan
		Rekomendasi Wisata Kuliner Jogja Terfavorit
		Tempat Penginapan Murah di Jogja Dekat Malioboro
		Tempat Wisata DI Yogyakarta
		Wisata Kuliner Yogyakarta
		Wisata Pantai di Jogja
6	Wisata Lombok	Tempat Wisata Di Lombok
		Wisata Kuliner Lombok
		Wisata Gili Trawangan, Pesona Bahari Lombok yang Menawan
		Inilah 10 Tempat Wisata di Lombok Paling Menawan
		WISATA INDONESIA , KUTA LOMBOK
		10 Tempat Wisata di Lombok yang Wajib Dikunjungi
		Wisata Gili Trawangan, Pulau di Lombok yang Seru dan Cantik
		Wisata Gunung Rinjani Lombok, Lanskap Eksotis dari Ketinggian
		10 Tempat Wisata Pantai di Lombok yang Wajib Dikunjungi
		Pantai Selong Belanak Lombok, Wisata Bahari Penuh Kedamaian
		Pesona Menakjubkan di Pantai Senggigi Lombok
		Pesona Gili Trawangan Lombok
		Keindahan Gili Meno Lombok
		Gili Air Lombok
		Menikmati Sunset di Pantai Senggigi Lombok

No	Query	Artikel Relevan
7	Wisata Bahari Indonesia	Lanskap Indah Pantai Kuta Lombok dari Bukit Mandalika
		Wisata Gunung Rinjani
		Wisata Karimunjawa, Satu Destinasi Bahari Menawan di Indonesia
		Wisata Bahari Lamongan
		Kepulauan Derawan, Pesona Wisata Bahari Kaltim
		Wisata Gili Trawangan, Pesona Bahari Lombok yang Menawan
		Wisata Bahari Lamongan, Destinasi Rekreasi Favorit di Jawa Timur
		Wisata Tanjung Lesung, Destinasi Bahari Banten yang Memikat
		Pantai Selong Belanak Lombok, Wisata Bahari Penuh Kedamaian
		Kepulauan Alor, Wisata Bahari NTT
		Wisata Bunaken, Pesona Taman Laut Manado Terpopuler
		Wisata kepulauan Seribu
		Diving Raja Ampat, Surganya Wisata Laut Indonesia
		Pantai Baron Gunungkidul, Kuliner Bahari Jogja Paling Menarik
		Taman Nasional Laut Kepulauan Seribu
		Pantai Batu Karas di Ciamis, Satu Destinasi Menarik Bagi Peselancar
8	Pantai Bali	Pantai Sanur Bali
		pantai kute bali
		Pantai Dreamland Bali, Destinasi Pantai yang Eksotis dan Alami

No	Query	Artikel Relevan
		Pantai Pandawa Nusa Dua Bali - Nuansa Indah Tersembunyi di Pulau Bali
		Pantai Kuta Bali, dari Romantisme yang Kental hingga Berselancar
		Pantai Lovina Bali
		Pantai Pandawa Bali, Nuansa Eksotis Penuh Ketenangan
		7 Tempat Wisata Pantai di Bali yang Wajib Dikunjungi
		Menikmati Indahnya Pantai Suluban
		Dreamland bali
		PANTAI KUTA
		Tanjung Benoa Bali
		Nusa Penida Bali
9	Objek Wisata Jakarta	Inilah 20 Tempat Wisata di Jakarta Paling Populer Dikunjungi
		Wisata Kota Tua Jakarta, Ini 5 Landmark Historis!
		4 Tempat Wisata Pantai di Jakarta Terpopuler
		Ini Pesona Wisata Pulau Tidung di Jakarta
		Ini 8 Restoran di Jakarta Paling Favorit
		Bernostalgia di Tempat Wisata Malam di Jakarta
		Wisata Kota Tua, Menelusuri Cikal Bakal Jakarta
		Tempat wisata di jakarta
		Museum Tekstil Jakarta
		Wisata Pantai di Jakarta
		Wisata Kuliner Jakarta Pusat
		Wisata Kuliner Jakarta

No	Query	Artikel Relevan
		Inilah 4 Tempat Wisata Kuliner Paling Unik di Jakarta Barat
		8 Tempat Wisata Kuliner di Jakarta Selatan dengan Menu Tradisional
		15 Tempat Wisata Kuliner di Jakarta yang Wajib Dikunjungi
		12 Tempat Wisata di Sekitar Jakarta yang Wajib Dikunjungi
		15 Mall di Jakarta yang Wajib Dikunjungi
10	Objek Wisata Jawa Timur	Wisata Terbaik di Jawa Timur
		10 Tempat Wisata Alam di Jawa Timur yang Wajib Dikunjungi
		Kota Wisata Batu Malang, Satu yang Terfavorit di Jawa Timur
		Wisata Bahari Lamongan, Destinasi Rekreasi Favorit di Jawa Timur
		Wisata Terbaik Jawa Timur
		10 Pantai Eksotis di Jawa Timur yang Wajib Dikunjungi
		objek wisata di malang
		7 Objek Wisata Lumajang Terpopuler
		Kerajinan Tangan Jawa Timur
		10 Tempat Wisata di Malang yang Wajib Dikunjungi
		Air Terjun Madakaripura, Meditasi Terakhir Sang Legenda di Jawa Timur
		Tempat Wisata Probolinggo
		Tempat Wisata Di Tuban
		6 Spot Wisata Jatim Park yang Mendidik dan Menghibur
		Wisata Blitar
		Tempat Wisata di Banyuwangi

No	Query	Artikel Relevan
		Tempat Wisata di Jember
		Mengunjungi Jawa Timur: Wisata Ponorogo
		Wisata Jatim Park Malang, Konsep Bermain Nan Edukatif
		Pantai Tambakrejo Blitar, Satu Pilihan Destinasi Wisata yang Menarik
		Wisata Gunung Bromo
		Pantai Popoh Tulungagung, Keindahan dan Kuliner Lautnya yang Lezat
		Pantai Pasir Putih Situbondo, Salah Satu yang Populer di Jatim
11	Objek Wisata Jawa Barat	Mengunjungi Objek Wisata di Kuningan Jawa Barat
		Tempat Wisata di Jawa Barat dan Sekitarnya
		Objek Wisata Bandung
		Mengunjungi Tempat Wisata di Sukabumi Jawa Barat
		Objek Wisata di Ciwidey Bandung
		Inilah 8 Tempat Wisata di Cirebon Paling Populer
		Wisata Tanjung Lesung
		Melepas Penat Dengan Mengunjungi Tempat Wisata Pelabuhan Ratu
		Tempat Wisata Ciwidey Bandung Selatan
		Inilah 20 Tempat Wisata di Bandung Paling Terkenal Untuk Dikunjungi
		Kunjungi 5 Tempat Wisata di Pangandaran yang Terkenal
		Wisata Ciater Subang, Tempat Pemandian Air Panas Alami

No	Query	Artikel Relevan
		Tempat Wisata di Puncak Bogor
		Wisata di Gunung Galunggung Tasikmalaya
		7 Pesona Memikat Tempat Wisata Ciwidey Bandung
		Tempat Wisata di Cirebon yang Menawan
		Green Canyon Pangandaran, dari Body Rafting hingga Alamnya yang Menawan
		Wisata Ujung Genteng, Satu Tujuan Rekreasi di Sukabumi yang Memikat
		Taman Safari Bogor, Destinasi Wisata di Habitat Alami Satwa
		Trans Studio Bandung, Satu Pilihan Rekreasi Wahana Kelas Dunia
		Pesona Lembang Bandung di Gunung Tangkuban Perahu
		Kampung Gajah Wonderland, Satu Destinasi di Bandung yang Favorit
		Wisata Gunung Pancar, Destinasi Akhir Pekan di Bogor yang Disukai

BAB VI

KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan yang diperoleh selama pengerjaan tugas akhir ini. Selain itu, juga terdapat beberapa saran terhadap tugas akhir ini yang diharapkan bisa membuat tugas akhir ini menjadi lebih baik lagi.

6.1. Kesimpulan

Kesimpulan yang diperoleh berdasarkan uji coba dan evaluasi yang telah dilakukan pada tugas akhir antara lain:

- Penghapusan parameter *norm* tidak memberikan perubahan yang signifikan terhadap akurasi hasil pencarian.
- Penggunaan FLC *query* tidak terlalu berpengaruh terhadap akurasi hasil pencarian.
- Penggunaan seluruh *rule fuzzy* dan *rule* yang sudah digabungkan memberikan hasil yang relatif sama.
- Pembobotan artikel dengan menggunakan pendekatan *fuzzy* meningkatkan hasil akurasi pencarian secara signifikan jika dibandingkan dengan metode *non fuzzy*.

6.2. Saran

Terdapat beberapa saran terkait tugas akhir ini yang diharapkan bisa membuat tugas akhir ini menjadi lebih baik. Saran-saran tersebut antara lain:

- Data artikel diperbanyak karena data artikel yang ada belum mencakup keseluruhan topik pariwisata.
- Penerapan *named entity recognition* untuk melakukan pengenalan nama daerah (Jawa Timur, Jawa Barat, dll) dan nama tempat (Candi Borobudur, Pantai Kenjeran, dll).

- Penerapan *auto correction* untuk antisipasi kesalahan memasukkan *query* dengan menggunakan algoritma *edit distance*.

DAFTAR PUSTAKA

- [1] B. Dictionary, "What is categorical data? Definition and meaning," Business Dictionary, [Online]. Available: <http://www.businessdictionary.com/definition/categorical-data.html>. [Accessed 7 April 2016].
- [2] D. Tubbs, "What is Flipboard and how to use it to build content," meersocial, 21 May 2014. [Online]. Available: <http://www.meersocial.com/2014/05/what-is-flipboard-and-how-to-use-it-to-build-content/>. [Accessed 7 April 2016].
- [3] B. Jensen and M. McNeese, "Evaluating the effectiveness and patterns of interactions with automated assistance in II systems," *Journal of the American Society for Information Science and Technology*, vol. 14, no. 56, pp. 1480-1503, 2004.
- [4] S. Robertson, S. Walker and M. Beaulieu, "Okapi-BM25 and TREC-7: Automatic ad hoc, filtering, VLC and filtering tracks," in *Proceedings of the seventh text retrieval conference*, 1999.
- [5] M. Rouse, "What is data preprocessing?," Techtarget September 2005. [Online]. Available: <http://searchsqlserver.techtarget.com/definition/data-preprocessing>. [Accessed 7 April 2016].
- [6] Import.io, "FAQs - Import.io," Import.io, [Online]. Available: <https://www.import.io/help/faqs/>. [Accessed 7 April 2016].
- [7] Google, "google/gson: A Java serialization/deserialization library that can convert Java Objects into JSON and back., Google, [Online]. Available: <https://github.com/google/gson> [Accessed 7 April 2016].
- [8] H. Wu, R. Luk, K. Wong and K. Kwok, "Interpreting TF-IDF term weights as making relevance decisions," *ACM Transactions on Information Systems*, vol. 3, no. 26, 2008.
- [9] Elastic, "Theory Behind Relevance Scoring," Elastic, [Online]. Available:

- <https://www.elastic.co/guide/en/elasticsearch/guide/current/scoring-theory.html>. [Accessed 18 April 2016].
- [10 E. Horstkotte, "Fuzzy Logic Overview," Togai InfraLogic [Online]. Available: <http://www.austinlinks.com/Fuzzy/overview.html>. [Accessed 17 May 2016].
- [11 doc.ic.ac.uk, "Fuzzy Rules and Fuzzy Control," [Online]. Available: http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/sbaa/report.fuzrules.html. [Accessed 19 April 2016].
- [12 Y. Gupta, A. Saini and A. Saxena, "A new fuzzy logic base ranking function for efficient Information," *Expert System with Applications*, vol. 42, no. 2015, pp. 1223-1234, 2014.
- [13 T. Point, "What are Web Services," Tutorials Point, [Online]. Available: http://www.tutorialspoint.com/webservices/what_are_web_services.htm. [Accessed 10 April 2016].
- [14 Netbeans, "Netbeans IDE - Databases," Netbeans, [Online]. Available: <https://netbeans.org/features/ide/database.html>. [Accessed 10 April 2016].
- [15 O. H. Alliance, "Android Overview," Open Handset Alliance [Online]. Available: http://www.openhandsetalliance.com/android_overview.html. [Accessed 19 April 2016].
- [16 codepath, "Using Android Studio," Github, [Online]. Available: https://github.com/codepath/android_guides/wiki/Using-Android-Studio. [Accessed 19 April 2016].
- [17 Simpligility, "ksoap2-android," Github, 15 October 2014 [Online]. Available: <http://simpligility.github.io/ksoap2-android/index.html>. [Accessed 19 April 2016].

BIODATA PENULIS



Marvin Zeson Abilo, lahir di Surabaya pada 11 Juni 1995. Ia merupakan anak sulung dari empat bersaudara. Jenjang pendidikan formal dimulai di SD Hang Tuah 10, SMPN 1 Sidoarjo, SMAN 1 Sidoarjo. Saat ini ia tercatat sebagai mahasiswa Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi 10 Nopember. Penulis sangat terkesan untuk belajar berbagai hal baru untuk menambah wawasan penulis.

Penulis dalam menyelesaikan pendidikan S1 mengambil rumpun mata kuliah (RMK) Komputasi Cerdas dan Visi serta memiliki ketertarikan di bidang Basis Data, Pemrograman Android, Data Mining, serta Machine Learning. Penulis dapat dihubungi melalui surel: marvin.zeson@gmail.com.